Universidad Carlos III de Madrid

Escuela Politécnica Superior

Grado en Ingeniería Informática

Trabajo Fin de Grado

2015

Desarrollo e Implementación de una Herramienta de Análisis de Secuencias de Acciones

Autor: Georgi Georgiev Spasov

Tutores: José Antonio Iglesias Martínez y Agapito Ismael Ledezma Espino

Agradecimientos

Quiero dedicar este documento a toda mi familia, que me han apoyado durante toda la vida en todo lo que me he propuesto. Su dedicación hacia mis estudios me ha permitido convertirme en una persona culta, simpática y lo más importante, realizada en la vida.

En cuanto a la gente de la Universidad, me gustaría agradecer a todos mis compañeros las increíbles tardes de risa haciendo prácticas. Será el mejor de los recuerdos que me llevo de estos años que hemos pasado juntos y nunca lo olvidaré.

En especial quiero hacer mención a mis compañeros más directos y con los que he pasado la mayoría del tiempo trabajando incesantemente en las tareas de la Universidad: Álvaro Maroto y Carlos Manzano.

A Álvaro, Maroto para los amigos, agradecerle que me haya aguantado durante todos estos años como compañero de prácticas y como apoyo durante las épocas duras de esfuerzo. Esas tardes "haciendo prácticas" fueron inolvidables y fuiste el mejor compañero que pude haber tenido en la carrera.

A Carlos, darle las gracias por ser el mejor amigo y compañero que podría tener. Tu apoyo y tu amistad son irremplazables. Gracias por ayudarme en la Universidad y fuera de ella, donde tu increíble personalidad y esfuerzo me han ayudado a superar estos difíciles años. Eres un verdadero amigo.

Gracias de nuevo a todos los que me habéis apoyado y no he mencionado aquí.

Resumen

Este documento presenta el Trabajo de Fin de Grado que he realizado en la Universidad Carlos III de Madrid, en la Escuela Politécnica de Leganés.

El proyecto que se ha desarrollado tiene como objetivo la automatización de la creación de una estructura arbórea (*trie*) que permite almacenar y analizar secuencias de acciones en dominio de datos secuenciales.

En el documento se describe toda la realización y estructura del software implementado, así como la experimentación que se ha realizado como demostración del funcionamiento del mismo. Gracias a este software se permite de forma automática crear tries y analizarlos, con un enfoque gráfico y estadístico.

La utilización del trie como estructura de representación de datos permite utilizar un enfoque estadístico para el análisis de secuencias de datos y la extracción de patrones. El software diseñado es genérico, por lo que puede ser utilizado en cualquier dominio, facilitando enormemente la labor de los investigadores.

Abstract

This document presents the Final Degree Work that I have done for the University Carlos III de Madrid, in the Polytechnic School of Leganés.

The developed project is aimed to the automatic creation of a tree structure (*trie*) that allows to store and to analyze sequences of actions in a sequential data domain.

In the document is described all of the work done and the structure of the implemented software, as well as the performed experimentation as demonstration of the functionality of the software system. Thanks to this software, automatic creation and analysis of tries are done with a statistical and graphical approach.

The usage of tries as structure to represent data allows to create a statistical approach to analyze data sequences and to perform pattern recognition. The design software is generic, so it can be used in any domain, facilitating the work of researchers.

Índice de contenido

Índice de ilustraciones

Índice de tablas

Índice de ecuaciones

Capítulo 1: Introducción

1.1. Motivación

El modelado de agentes se define como *la capacidad de adquirir, inferir y almacenar el conocimiento (comportamiento, creencias, metas, acciones, planes…) de otros agentes* [1]. La finalidad del modelado de agentes es inferir lo que está haciendo o pretende hacer un agente en base a las acciones que realiza o los eventos que ocurren en su entorno. Se considera como agente, un ser humano, un sistema software o un sistema hardware.

Cuando se considera como agente un ser humano, existe un amplio campo de investigación sobre el estudio de su comportamiento. Un área de interés en el estudio del comportamiento humano es la comercialización o marketing. Conocer los clientes y sus preferencias respecto a un determinado producto es un tema de gran interés en esta área, donde se busca incrementar la demanda de un producto dirigido a cierto tipo de consumidores. Un ejemplo de la aplicación del estudio del comportamiento humano en esta área se puede encontrar en empresas de comercio como Amazon, que utiliza este tipo de técnicas para analizar a sus clientes y hacer predicciones, en base a sus comportamientos, de productos que potencialmente podrían ser comprados por estos usuarios. De esta manera se ha creado un sistema de recomendaciones que genera importantes ingresos en el modelo de negocio que siguen empresas como Amazon [2].

En los últimos años, gracias a los avances en la robótica y en la inteligencia artificial, existen cada vez más sistemas inteligentes que pueden comunicarse con una persona. Estos sistemas se comunican con personas mediante interfaces gráficas, sonidos e incluso utilizando extensiones de su cuerpo físico, en el caso de los robots. Construir sistemas amigables y funcionales que puedan interactuar con cualquier usuario, independientemente de su formación educativa y especializada es un tema que está actualmente en investigación y poco a poco surgen prototipos que intentan emular por completo el comportamiento humano. Un ejemplo de ello son los asistentes robot. Ejemplos de este tipo de agentes son NAO [3] y ASIMO [4]. En el caso de ASIMO, es capaz de ver, hablar, levantar objetos, correr y subir escaleras. La comunicación con éste se puede realizar mediante el habla, por lo que el entendimiento del habla forma parte clave para la comunicación con el robot.

La construcción de sistemas basados en el comportamiento de un agente se encuentra en otras muchas áreas, como en la industria de los videojuegos. En esta área, el sistema inteligente del videojuego debe adaptarse al comportamiento del jugador, para ofrecer una experiencia de desafío a la inteligencia y habilidad que manifiesta el jugador. Un ejemplo de ello se puede encontrar en [5], donde se pretende modelizar el comportamiento de los jugadores en el juego de *World Of Warcraft*.

Con los avances tecnológicos actuales y la expansión mundial de Internet, la seguridad y protección de los sistemas informáticos se ha vuelto una prioridad. Los ataques informáticos que pretenden invadir la privacidad o dañar un sistema son un tema prioritario, tanto para usuarios corrientes, empresas e incluso gobiernos. El estudio del comportamiento de los agentes maliciosos presentes en estos ataques contribuye a la creación de defensas, tanto software, como hardware contra este tipo de ataques. Existe una amplia bibliografía que trata el comportamiento de un agente malicioso dentro de un sistema, como por ejemplo en [6], donde se utiliza el modelado de comportamiento de un software malicioso para identificar sus componentes.

La motivación de este proyecto reside en aportar al modelado de agentes un sistema automático que facilite el análisis de comportamientos secuenciales.

1.2. Objetivos

En esta sección se describen los objetivos, tanto general, como específicos que el proyecto tiene como meta alcanzar.

1.2.1. Objetivo general

El objetivo general de este proyecto es el desarrollo de una herramienta software independiente del dominio, para el análisis de secuencias de acciones, utilizando como método de representación una estructura de árbol (trie).

1.2.2. Objetivos específicos

Para lograr el objetivo general de este proyecto, se deben lograr los siguientes objetivos específicos:

- Dotar al sistema software de una interfaz gráfica, donde el analista pueda indicar todos los parámetros de configuración de la formación y representación gráfica del trie.
- Diseñar el sistema para que sea capaz de operar con independencia del dominio de datos, pudiendo interpretar cualquier secuencia de acciones o eventos.
- Obtener diversas métricas estadísticas enfocadas al análisis de secuencias de acciones y extracción de patrones. Estas métricas deben ser:
    - Valores de soporte.
    - Eventos del dominio.
    - Filtrado de acciones con frecuencias de inserción en el trie un con umbral mínimo.
    - Filtrado de acciones con frecuencias estadísticas en el trie con un umbral mínimo.
- Representar gráficamente el trie resultante de la configuración impuesta por el analista que utiliza la herramienta.

1.3. Estructura del documento

En esta sección se describe la estructura que presenta el documento:

Capítulo 1: se presenta la motivación para realizar este proyecto como Trabajo Fin de Grado. Además, se describe el objetivo general del proyecto, así como los objetivos específicos necesarios para lograr el objetivo general.

Capítulo 2: esta sección engloba el estado del arte en el que se enmarca esta investigación, definiendo conceptos y mostrando investigaciones más relevantes relacionadas con los objetivos de este proyecto.

Capítulo 3: en esta sección se realiza una descripción detallada del sistema implementado, repasando todos los flujos de ejecución y restricciones bajo los que el sistema opera. Además, se describen las tecnologías utilizadas y los motivos por los que se han seleccionado para la elaboración de este proyecto.

Capítulo 4: representa la experimentación realizada con el sistema implementado. En la sección se detalla el dominio y los datos utilizados para verificar el funcionamiento y la utilidad del sistema. Además, se presentan los resultados obtenidos de la experimentación.

Capítulo 5: esta sección trata sobre el desarrollo del proyecto, exponiendo la planificación que se ha seguido y el presupuesto necesario para llevar a cabo el sistema a nivel comercial.

Capítulo 6: se exponen las conclusiones de la experimentación y de la realización del proyecto. Además, se describen los trabajos futuros que se pueden hacer en base a la investigación llevada en este proyecto.

Anexos: la última parte del documento se compone de una colección de anexos. En el primer anexo se presentan los resultados de la experimentación omitidos en el Capítulo 4. En el segundo anexo, se presenta el Manual de Usuario que describe la estructura, la ejecución y el tratamiento de errores que posee la herramienta desarrollada. Por último, los anexos tres, cuatro y cinco contienen las traducciones en Inglés de la introducción, los resultados de la experimentación y las conclusiones y trabajos futuros.

Capítulo 2: Estado del arte

En este capítulo se presentan las áreas de investigación relacionadas con el desarrollo de este proyecto. Se trata del estudio del reconocimiento de actividades, el modelado de usuarios, la minería de secuencias y el modelado automático de agentes.

Todos estos temas están directamente relacionados con la investigación que se presenta, ya que forman los cimientos teóricos sobre los que se basa la realización de este proyecto.

2.1. Reconocimiento de actividades

A finales de los años 70 y en la década de los 80, se publicaron diversas investigaciones orientadas al reconocimiento de actividades, bajo el nombre de Reconocimiento de Planes. En 1978, Schmidt et al. [7] presentaron una investigación pionera en este campo. El objetivo de esta investigación era determinar el entendimiento del ser humano de las acciones de otros seres humanos. En 1986, Kautz y Allen [8] definen el Reconocimiento de Planes como *el problema de identificar un conjunto mínimo de acciones de alto nivel suficientes para explicar el conjunto de acciones observadas*. Entre los años 1980 y 2000, el Reconocimiento de Planes se abordó desde tres enfoques:

- Reconocimiento de planes basado en observación [9]
- Reconocimiento de planes basado en análisis sintáctico [10]
- Reconocimiento de planes basado en inferencia similar [11]

Otra área de investigación en la que se basa el reconocimiento de actividades es el Modelado de Usuarios. Según Zukerman y Albrecht [12], *el Modelado de Usuarios implica inferir información no observable sobre un usuario a partir de información observable sobre él / ella, por ejemplo, su / sus acciones o declaraciones*.

Desde los años 90, con el surgimiento y expansión de Internet, las redes sociales y el comercio electrónico, se han llevado a cabo numerosas investigaciones sobre lo que hacen los otros. Han surgido nuevos enfoques de esta disciplina, como el Reconocimiento de actividad basado en sensores [13] y el Reconocimiento de actividad basado en visión [14].

El reconocimiento de actividades es una línea de investigación con diversos orígenes, por lo que se puede definir desde diferentes puntos de vista. A pesar de ello, esta disciplina trata de determinar lo que están haciendo los otros, sean personas, robots o agentes software.

2.1.1. Procedimiento

Según Chen et al [13], el reconocimiento de actividades es una tarea compleja, pero se puede formalizar en cuatro tareas básicas:

- Captura de actividades: se trata de elegir y desplegar el conjunto de sensores en el entorno con el fin de monitorizar y capturar el comportamiento de un agente.
- Almacenaje y procesamiento: esta tarea engloba la recolecta, el almacenamiento y el procesamiento mediante diferentes técnicas de la información recibida. En esta tarea es vital reconocer el tipo de acción que se está observando: una acción en concreto, un plan de acciones o un estado general.
- Modelos de actividad: esta tarea tiene como objetivo crear modelos de actividad de un agente, basándose en la información obtenida por la interacción agente-agente.

- Algoritmia de razonamiento: la última tarea tiene como objetivo seleccionar y desarrollar algoritmos de razonamiento que permitan deducir las actividades que realiza un agente en base a la información obtenida de su entorno operacional.

2.1.2. Criterios de modelado

En la actualidad existen numerosos enfoques en el área del Reconocimiento de actividades. En términos generales, los criterios de clasificación de éstos pueden ser agrupados como: sujetos de reconocimiento, sujeto reconocedor, datos y modelo.

2.1.2.1. Sujeto de reconocimiento

Los sujetos de reconocimiento se consideran aquellos a los que se les aplicará el proceso de modelado. Como sujetos de reconocimiento se podrían tomar como sujetos personas, robots o agentes software. En el caso de ser personas, se podría modelar, por ejemplo, el comportamiento de una persona mayor o con discapacidad [15] o el comportamiento de un estudiante que interactúa con una plataforma de aprendizaje [16].

En caso de que el sujeto de reconocimiento fuera un robot, se podría modelar, por ejemplo, el comportamiento de robots que juegan al fútbol. Este caso en particular ha despertado un gran interés en la comunidad científica (RoboCup [17]).

Para el caso de que el sujeto de reconocimiento fuera un agente software existen muchos ejemplos. Uno de ellos está relacionado con la RoboCup, donde se intenta modelar el comportamiento de agentes software que juegan al fútbol simulado [18].

2.1.2.2. Sujeto reconocedor

El sujeto reconocedor es aquel que modela el comportamiento de otro agente. Lo más habitual es que esta tarea la lleve a cabo un sistema informático desde un ordenador, pero existen otros tipos de sujetos reconocedores. Un ejemplo de ellos se puede encontrar en [18] [19], donde agentes software autónomos son los sujetos reconocedores en un partido simulado de la RoboCup.

Por otro lado, el sujeto reconocedor puede ser un robot, que modela el comportamiento de otros robots. Es el caso de [20], donde un robot interacciona con otro robot y modela su comportamiento.

2.1.2.3. Datos

Este enfoque se centra en la definición y la selección de los datos con los que se va a realizar el reconocimiento de actividades. Actualmente, en función del método de captura de datos, las investigaciones suelen clasificarse en Reconocimiento basado en sensores y Reconocimiento basado en visión.

2.1.2.4. Modelo

Actualmente, en función del proceso de creación de los modelos de actividad, existen dos tipos de modelos: modelo orientado a datos y modelo orientado a conocimiento [13].

El modelo orientado a datos utiliza técnicas de minería de datos y aprendizaje automático para generar los modelos de actividad en base a bases de datos preexistentes. Este tipo de modelos puede clasificarse como generativo [21] y discriminativo [22]. Se tratan de modelos generativos,

ya que intentan construir una descripción completa del espacio de entradas. Por otra parte, son discriminativos debido a que únicamente modelan la correspondencia entre los datos de entrada y los datos de salida.

2.1.3. Enfoques actuales

A continuación se presentan los enfoques actuales en los que se pueden agrupar los procesos de reconocimiento de actividades.

2.1.3.1. Reconocimiento basado en visión

Este enfoque utiliza la percepción visual como método de modelado de actividades. Hace uso de videocámaras para capturar el comportamiento de un agente y los cambios que se producen en su entorno operacional [14].

Este tipo de reconocimiento utiliza técnicas de visión por computadora para analizar los datos obtenidos por las videocámaras para detectar patrones de comportamiento.

Con la gran cantidad de videocámaras dispuestas en espacios públicos, las investigaciones y aplicaciones de este tipo de reconocimiento están a la orden del día. Los avances tecnológicos en videocámaras y sus sensores han expandido este campo, donde se utilizan una o varias cámaras en estéreo, cámaras infrarrojas e incluso cámaras RGBD de bajo coste, como la Kinect de Microsoft para el reconocimiento de actividades de personas [23].

En cuanto a los escenarios de captura, se intenta modelar el comportamiento de uno o varios agentes mediante las interacciones de estos con los objetos de su entorno.

Los métodos más destacados utilizados en el Reconocimiento basado en visión son los filtros de Kalman, los Modelos Ocultos de Markov y las técnicas de flujo óptico.

El procedimiento más común de este enfoque de reconocimiento de actividades es el siguiente [13]:

- Detección de objetos
- Seguimiento de comportamiento
- Reconocimiento de actividad
- Evaluación de la actividad de alto nivel

Como referencia de este tipo de reconocimiento de actividades, se puede consultar [14], donde se realiza de forma exhaustiva los progresos de este enfoque en el reconocimiento de actividad de personas.

2.1.3.2. Reconocimiento basado en sensores

Con el auge tecnológico producido recientemente, y el interés cada vez mayor de la computación centrada en personas, existen un gran número de sensores, que entre otras cosas, se utilizan para el reconocimiento de actividades. En el mercado se pueden encontrar sensores de contacto, sensores de identificación por radiofrecuencia o RFID por sus siglas en Inglés, acelerómetros, detectores de movimiento, etc.

Existen diversos criterios de agrupación de los sensores, pero enfocado al reconocimiento de actividades, se pueden considerar estos: sensores portátiles y densidad de sensores.

El Reconocimiento de actividad basado en sensores portátiles utiliza sensores colocados directa (mediante un cinturón, adhesivo, etc.) o indirectamente (incrustados en la ropa, complementos, etc.) en el cuerpo de una persona. Este tipo de sensores monitoriza el estado y la actividad de la persona, como la posición del cuerpo, su temperatura, el movimiento que realiza o el ritmo cardíaco [24, 25].

El Reconocimiento de actividad basado en la densidad de sensores se basa en la idea de que se puede modelar una actividad como la interacción entre un agente y un objeto. En este enfoque se utilizan sensores unidos a objetos, de forma que se recogen los datos de los sensores para establecer la relación agente-objeto que define una actividad. Este tipo de reconocimiento de actividades está muy ligado a al desarrollo de aplicaciones para la Vida Cotidiana Asistida por el Entorno o *AAL* por sus siglas en Inglés. Este tipo de enfoque se ha utilizado en distintas disciplinas, como la medicina [26] y los entornos inteligentes para niños [27].

2.1.3.3. Reconocimiento basado en interacción

En este enfoque se engloban las investigaciones que se realizan entorno a la interacción entre un agente que se modela y el sujeto que lo modela a través de sistemas informáticos.

En primer lugar existe una línea de investigación que trata de modelar el comportamiento de agentes software y otros sistemas informáticos en entornos virtuales. Un ejemplo de ello es [18, 28], donde agentes software son capaces de modelar el comportamiento que presentan jugadores en partidos de fútbol virtuales. Otros ejemplos se pueden encontrar en juegos virtuales como el póker virtual [29] o las subastas [30].

Otra línea de investigación enmarcada en este enfoque es la orientada a modelar el comportamiento de usuarios que utilizan aplicaciones informáticas. Los datos de estas investigaciones provienen del uso que hace el usuario en el sistema informático, es decir, la interacción hombre-máquina, o lo que se conoce como Modelado de Usuario.

2.1.4. Aplicaciones

Teniendo en cuenta que esta disciplina es el punto de convergencia de muchas otras, las aplicaciones que tiene se aplican en diversas áreas.

- Salud: en esta área, el reconocimiento de actividades se utiliza para analizar y comprender las actividades que realizan los pacientes dentro de un entorno, sea un hospital, una residencia, etc. Los resultados obtenidos ayudan a crear un diagnóstico, asignar un tratamiento y crear un programa de dedicación a los pacientes. Con ello se facilita la labor del personal médico y la calidad de vida del paciente es mejorada gracias al aumento de la fiabilidad del diagnóstico, al tratamiento prescrito y los resultados del proceso de reconocimiento de actividades [14, 31, 32].
- Entornos inteligentes: esta área de aplicación tiene como objetivo mejorar la vida cotidiana de las personas, basándose en las interacciones que realizan con su entorno cotidiano. Enmarca tareas cotidianas como cocinar, conducir, estudiar, etc. Con los avances tecnológicos actuales, cada vez es más común el término “casas inteligentes”, en los que las tareas cotidianas son asistidas por sistemas que estudian el comportamiento de las personas con las que interaccionan [33]. Otra aplicación de entornos inteligentes se pueden encontrar en las denominadas “oficinas inteligentes” [34, 35]. Otro término que está en auge es el de “coche inteligente”, capaz de ser autónomo y de proporcionar ayudas a la conducción [36, 37].

- Entretenimiento: en esta área, el reconocimiento de actividades se utiliza como forma de mejorar el estilo de vida de las personas. El entretenimiento se ve puede ver como la práctica de deportes, el ocio o los cada vez más populares, videojuegos, entre otros. Un ejemplo de la aplicación de reconocimiento de actividad en el deporte se puede encontrar en [38] aplicada al tenis. Respecto a los videojuegos existe mucha literatura como la que se encuentra en [39, 40, 41], donde se describe el reconocimiento de agentes en base a sus actividades.
- Seguridad: en esta categoría se tienen todas aquellas aplicaciones enfocadas a la vigilancia y control de masas [42, 43, 44]. Además, se enmarcan también las investigaciones enfocadas a sistemas de detección de intrusos y malware [45, 46, 47, 48 49].
- Militar: en este ámbito, el reconocimiento de actividades es clave para incrementar la eficiencia, tanto de los soldados, como de las misiones que realizan. Como ejemplo de aplicación se pueden encontrar [50], donde se estudia la hipotermia en soldados y [51, 52], donde se expone el reconocimiento de tácticas militares.

2.2. Modelado de Usuario

Esta área está estrechamente relacionado con el reconocimiento de actividades. De forma general, el modelado de usuario se puede definir como la personalización y adaptación sistemas informáticos en base a las necesidades de un usuario. Para ello es necesario modelar un usuario mediante una representación del conocimiento que posee sobre un dominio. En [12] se definen el modelado de usuario como *la capacidad de deducir información oculta sobre un usuario en base a la información visible que éste expresa*.

En el ámbito de la investigación, el modelado de usuario está también muy relacionado con la interacción hombre-máquina, cuyos resultados buscan mejorar la experiencia de usuario, sea en el diseño de aplicaciones, en el ámbito de la psicología mediante las teorías cognitivas y de comportamiento o en la antropología, que estudia la interacción entre las personas y el trabajo.

2.2.1. Enfoques

En el surgimiento de esta disciplina, el enfoque que se seguía residía en el modelado de usuario mediante colecciones de intenciones o preferencias, denominadas bibliotecas de planes [53]. Este proceso se hacía manualmente, por lo que era complicado, y costoso. Con la era de Internet y los avances tecnológicos actuales se ha demostrado que no todos los usuarios interactúan de la misma forma con una máquina, por lo que estas bibliotecas han perdido precisión en sus conclusiones.

Actualmente, el modelado de usuario se realiza mediante técnicas de aprendizaje automático y métodos estadísticos.

Webb et. al [54] propone que, dependiendo del propósito de creación del modelo de usuario, existen diferentes capacidades que esta disciplina puede revelar:

- Procesos cognitivos subyacentes en las acciones de un usuario.
- Diferencias entre habilidades de un usuario y un experto.
- Patrones y preferencias de un usuario.
- Características de un usuario.

La mayor parte de las técnicas de aprendizaje automático se centran en las dos primeras capacidades. En cuanto a los métodos estadísticos, el modelado de usuario se centra en predecir las acciones futuras de un usuario en determinadas circunstancias. Para ellos se utilizan, entre otros, modelos de Markov, redes bayesianas y métodos basados en frecuencias, como TF-IDF (frecuencia de término – frecuencia inversa de documento, por sus siglas en Inglés).

2.2.2. Aplicaciones

En la actualidad, conocer lo que un usuario opina sobre un producto está siendo cada vez más importante para los fabricantes y comerciales. Tomando como referencia la mayor conferencia sobre modelado de usuarios y sistemas adaptativos UMAP (*User Modeling, Adaptation and Personalization*, por sus siglas en Inglés), las aplicaciones de esta disciplina se pueden clasificar como:

- Social: en esta área se engloba todo lo relacionado con la interacción entre individuos para conseguir un objetivo. Se puede hablar de modelado de individuos, análisis de redes sociales, sistemas de recomendaciones, aprendizaje social, *crowdsourcing*, etc.
- Big Data: en cuanto al análisis de grandes cantidades de datos, se puede aplicar a tareas como la personalización de un sistema a gran escala, seguimiento de la Web, la toma de decisiones complejas, etc.
- Computación ubicua: respecto a la integración de la informática en la vida de las personas de forma que no sea percibida como un concepto diferenciado, el modelado de usuarios se utiliza en diseño e interacción en redes de sensores, señales fisiológicas, interacción natural con un usuario (sea habla, gestos, etc.), diseño de dispositivos móviles y portátiles, etc.

2.3. Minería de secuencias

2.3.1. Definición

Todas las acciones que produce un agente son secuenciales, por lo que el estudio de la secuencialidad en las acciones es de vital interés para el modelado de agentes.

La minería de secuencias puede considerarse como una subdisciplina de la minería de datos, donde el principal objetivo es estudiar la estructura y secuencialidad de los elementos que componen cada secuencia de acciones.

Una secuencia, de acuerdo a la Real Academia Española es *una serie o sucesión de cosas que guardan entre sí cierta relación*. Una secuencia se compone de *n* elementos ordenados y su representación es la siguiente:

$$\text{Secuencia} = \{e_1, e_2, e_3, \ldots, e_n\}$$

La secuencia se compone de *n* elementos que están relacionados entre sí, ordenados en una dimensión, sea espacio o tiempo.

Un ejemplo de ordenación por espacio se puede encontrar en la formación de las cadenas de ADN, donde cada elemento de la secuencia es un nucleótido. Una secuencia ordenada en el tiempo se puede obtener, por ejemplo, con los comandos que un usuario ha introducido en una terminal durante cierto período de tiempo.

2.3.2. Enfoques

En función de la finalidad que se persigue, las tareas de la minería de secuencias se pueden enumerar como:

- Descubrimiento de subsecuencias relevantes: en esta tarea se pretende encontrar las secuencias más representativas de un agente para determinar un patrón en el comportamiento de un agente. Un ejemplo de esta finalidad se puede encontrar en [55], donde se presenta el algoritmo SEQUEST que realiza este tipo de tarea.
- Predicción de secuencias: esta tarea trata de encontrar el conjunto de secuencias futuras basándose en las secuencias precedentes. [56] expone un algoritmo de procesado de texto capaz de predecir secuencias.
- Clasificación de secuencias: en esta tarea se pretende construir un modelo clasificador a partir de un conjunto de secuencias etiquetadas, que se utilizan en la fase de entrenamiento del modelo. En [57], se presenta un modelo clasificador para formas y caras basado en la clasificación de secuencias mediante Modelos de Markov.
- Agrupación de secuencias: cuando se dispone de un conjunto de secuencias no etiquetadas, esta tarea pretende crear subconjuntos de secuencias que comparten características similares. Cada subconjunto de secuencias se denomina clúster. En [58] se propone un modelo de agrupación basado en valores estadísticos que miden la similitud entre secuencias.

2.3.3. Aplicaciones

En cuanto a aplicaciones, la minería de secuencias se utiliza en diversos campos:

- Genética: la formación de la estructura genética del ADN se puede modelar como una secuencia de nucleótidos, en el que su orden y frecuencia determinan la composición de un ente biológico. Existen numerosos trabajos sobre este tema. Un ejemplo de ello se pueden encontrar en [59], donde se presenta un método de clasificación de secuencias de ADN que determina si la secuencia analizada pertenece a la bacteria E.Coli, presente en el sistema digestivo humano.
- Detección de intrusos: la detección de intrusos en un sistema informático es un tema muy presente de la actualidad. Esta detección se puede realizar, entre otras técnicas, estudiando la secuencia de comandos que el sistema recibe. En [60], se propone un algoritmo que mide las similitudes entre secuencias para poder determinar usuarios legítimos e intrusos.
- Modelado automático de agentes: la observación del comportamiento de un agente permite realizar modelos de otros agentes basándose en el observado previamente. Las secuencias de acciones se descomponen y categorizan en acciones atómicas y mediante dichas acciones atómicas es posible crear un comportamiento. En [61], se propone el uso de aprendizaje autónomo no supervisado de secuencias con el objetivo de modelar el comportamiento de agentes basándose en las observaciones de los propios agentes.

2.4. Modelado automático de agentes

En esta sección se describen dos de los sistemas de modelado de agentes expuestos en la tesis doctoral *Modelado Automático del Comportamiento de Agentes Inteligentes* [62]. Estos sistemas forman las bases de la arquitectura del sistema implementado en este proyecto.

Es importante mencionar que los dos sistemas que se van a detallar en esta sección han sido modelados, entre otros dominios, para la propuesta de la competición RoboCup Coach 2006 [63].

2.4.1. Modelado de Agentes Mediante Comparación de Secuencias de Eventos: M-Comp

Este sistema tiene como objetivo modelar el comportamiento de un agente a partir de la diferencia obtenida tras una comparación de dos comportamientos diferentes. La diferencia de los dos modelos de comportamiento diferentes genera el modelo del agente. En la Ilustración 1 se muestra la arquitectura general de este sistema:

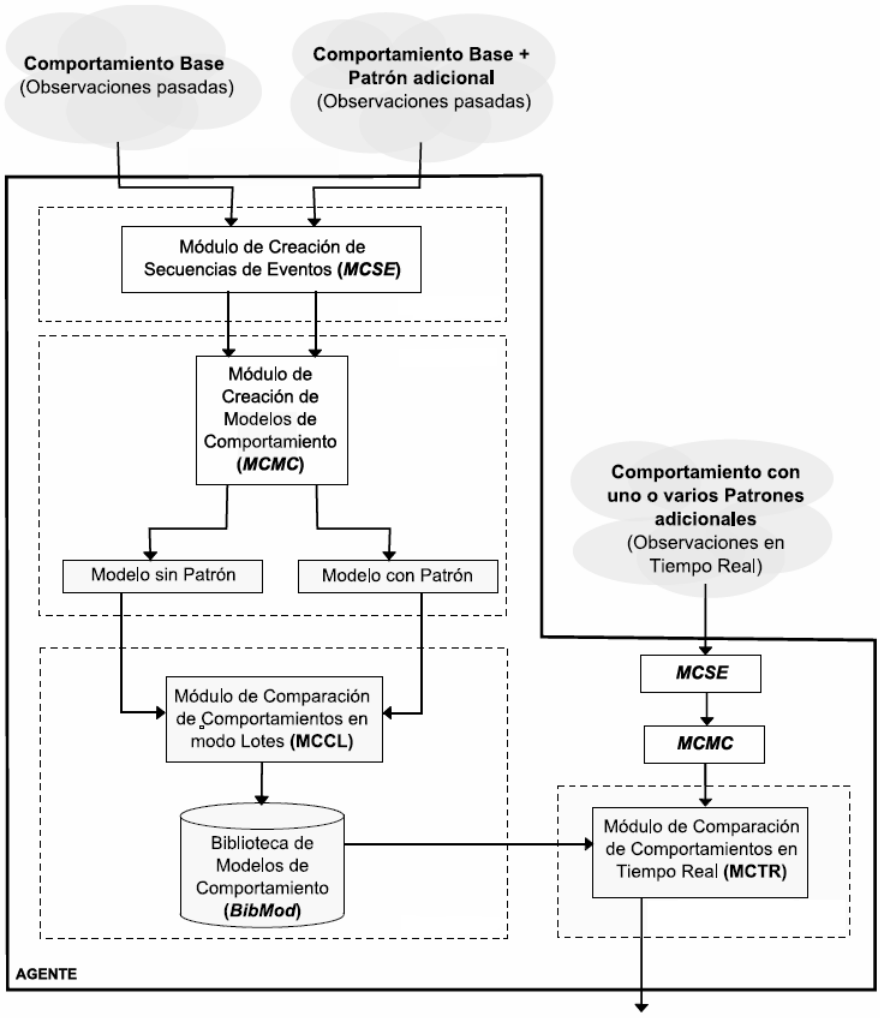

Ilustración 1: Sistema M-Comp [62].

A continuación se explican los módulos de este sistema en los que el desarrollo del proyecto se ha basado. Se trata del Módulo de Creación de Secuencias de Eventos (MCSE) y el Módulo de Creación de Modelos de Comportamiento (MCMC).

2.4.1.1. Módulo de Creación de Secuencias de Eventos (MCSE)

El objetivo de este módulo es obtener las secuencias de eventos o acciones provenientes de un conjunto de agentes.

El módulo se encarga de recibir y procesar las observaciones realizadas por el conjunto de agentes. En primer lugar, se debe especificar que de todo el conjunto de observaciones se deben extraer únicamente aquellos datos que permitan formar secuencias de eventos. Se define como evento aquella observación que ocurre en un espacio y tiempo determinados y que define unívocamente una acción específica de uno o varios agentes.

Una vez obtenidos los eventos se procede a crear la secuencia entre ellos. Es necesario recordar que una secuencia se forma por un conjunto de eventos contiguos y ordenados de la siguiente forma:

$$Secuencia = \{e_1, e_2, e_3, \ldots, e_n\}.$$

Se debe hacer hincapié en que la duración de los eventos es irrelevante, en este caso, y que su tratamiento se puede considerar como trabajo futuro de este proyecto.

Como salida, este módulo produce un conjunto de secuencias de eventos.

2.4.1.2. Módulo de Creación de Modelos de Comportamiento (MCMC)

Este módulo se encarga de crear el modelo de comportamiento en base al conjunto de secuencias previamente formado.

El enfoque, tanto de la tesis doctoral que se toma como base, como el de este proyecto, es que se puede construir un modelo de comportamiento basándose en la frecuencia de repetición de los eventos producidos por un agente. Los eventos más relevantes son aquellos que aparecen con mayor frecuencia. Esta idea es muy utilizada en las áreas de recuperación de información y la minería de textos, donde se utilizan estadísticas como TF-IDF [64] para evaluar la importancia de un término o documento.

A la hora de modelar el comportamiento de un agente es importante mantener la secuencialidad y temporalidad de los eventos que ha realizado. Tómese como ejemplo las siguientes secuencias de comandos UNIX y nótese la diferencia entre ellas:

rm a.txt; mv b.txt a.txt

mv b.txt a.txt; rm a.txt

Teniendo en cuenta estos dos aspectos, de mantener la secuencialidad y temporalidad de los eventos, así como de utilizar métricas estadísticas para representar la importancia de un evento, se ha decidido utilizar la estructura denominada *trie* para modelar el comportamiento de un agente.

El Trie es una estructura de datos que permite la indexación y recuperación de información de forma eficiente utilizando una estructura de tipo arbórea. Trie proviene como abreviatura de la palabra inglesa *Retrieval* y fue introducido como concepto por Edward Fredkin en su artículo "Trie Memory" publicado en 1960 [65].

Este tipo de estructura representa un caso especial de un Autómata Finito Determinista, donde cada nodo representa un estado y cada enlace (dirigido) representa una transición entre estados (nodo padre y nodos hijo).

La siguiente ilustración ilustra la forma y los elementos de un trie:

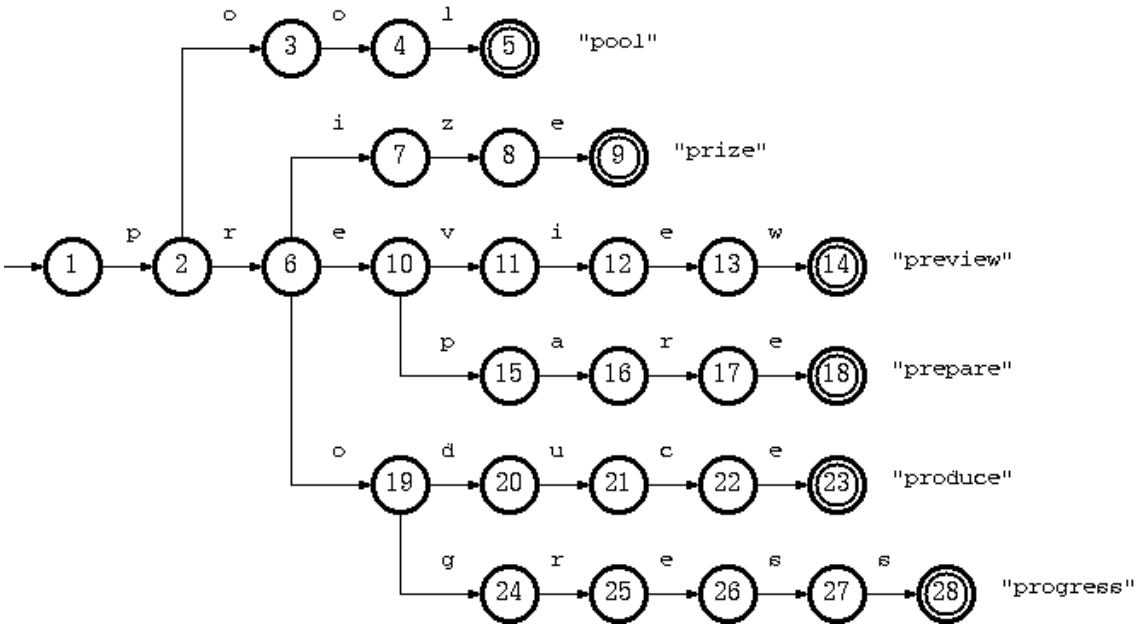

Ilustración 2: Estructura arbórea del trie [66].

La lectura del trie comienza siempre desde el nodo raíz (nodo 1 en la ilustración 2) y prosigue en dirección a las flechas. Cada transición representa la adición de un carácter a la cadena previamente formada. Este tipo de estructuras se utilizan para indexación y búsqueda de palabras o *strings*, ya que permiten aprovechar el concepto léxico de prefijo. Todos los descendientes de un nodo tienen como prefijo común el nodo padre.

En la ilustración 2 se ejemplifica la formación de las palabras “pool”, “prize”, “preview”, “prepare”, “produce” y “progress”. El nodo raíz del trie es el nodo 1 y mediante las sucesivas transiciones se forman las diferentes palabras.

En este proyecto se ha decidido aprovechar el concepto de trie aplicado a una sucesión de acciones o eventos. En el trie original, cada nodo y enlace representaban el estado y la transición de un carácter a la formación de una sucesión de caracteres o palabras. En este desarrollo cada nodo es una acción o evento y las transiciones entre nodos vienen definidas por un método de segmentación de secuencias de acciones (que se detallará más adelante).

El proceso de formación del trie se compone de dos fases: segmentación y almacenamiento.

Segmentación

En esta fase se extraen las acciones individuales del conjunto original de acciones y se establecen las relaciones entre éstas en base a un criterio de segmentación. En este proyecto se han utilizado dos criterios de segmentación: longitud de la secuencia de acciones y palabras clave que denotan el inicio y el final de cada secuencia de acciones. A continuación se propone un conjunto de acciones individuales de ejemplo para ilustrar el proceso de formación de las secuencias de acciones. Supóngase que el conjunto de acciones está formado por las acciones {*ls, date, ls, date, cat*}. Supóngase que se realiza el método de segmentación por longitud de secuencia y se elige una longitud de tres. El conjunto de secuencias de acciones sería el siguiente: {*ls, date, ls*}, {*date, ls, date*} y {*ls, date, cat*}.

Almacenamiento

Una vez obtenidas las secuencias de acciones es el momento de insertarlas dentro del trie. El trie comienza con el nodo raíz, común a todas las secuencias de acciones. Cada acción individual es encapsulada como un nodo y entre corchetes se representa la frecuencia con la que la acción se ha insertado en el trie (se comienza en uno y cada vez que se inserta una acción en la misma posición del trie, este contador se incrementa en uno). Se lee la primera secuencia de acciones y se representa como una rama:

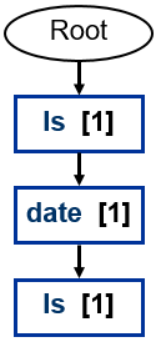

Ilustración 3: Formación del trie (1).

A continuación se leen los sufijos o nodos hijo de cada acción de dicha secuencia, es decir, se crean las ramas de los nodos hijo de la primera secuencia de acciones. Estas ramas vienen representadas por las subsecuencias {*date, ls*} y {*ls*}:

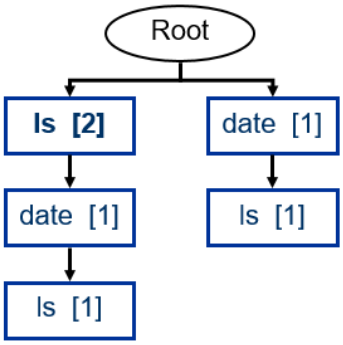

Ilustración 4: Formación del trie (2).

Nótese que el nodo {*ls*} ya ha sido insertado previamente, por lo que su frecuencia de inserción aumenta, pasando de ser uno a ser dos. A continuación se repite este proceso hasta que todas las secuencias de acciones y sus nodos han sido procesados. El trie resultante es el siguiente:

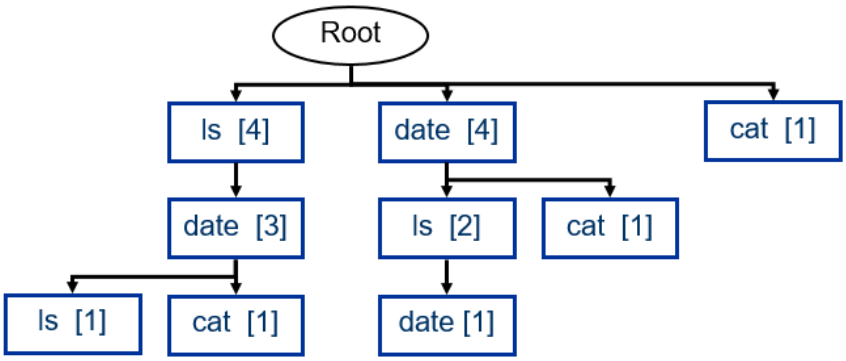

Ilustración 5: Formación del trie (3).

Una vez formado el trie, el siguiente paso es calcular la relevancia de cada evento. Para ello, en este sistema se ha utilizado la métrica estadística denominada Chi-cuadrado. Esta métrica determina la relevancia de un término en función de los eventos observados y los eventos esperados.

En primer lugar se crea una tabla de contingencia que representa la probabilidad de que dos eventos estén relacionados. La siguiente ilustración representa la estructura de esta tabla:

	Evento	Prefijo diferente	Total
Prefijo	O_{11}	O_{12}	$O_{11} + O_{12}$
Prefijo diferente	O_{21}	O_{22}	$O_{21} + O_{22}$
Total	$O_{11} + O_{21}$	$O_{12} + O_{22}$	$O_{11} + O_{12} + O_{21} + O_{22}$

Tabla 1: Tabla de cálculo de Chi-cuadrado [62].

Los componentes de esta tabla son los siguientes:

- O_{11}: representa el número de veces que el evento se ha insertado en el trie, dentro del mismo nodo.
- O_{12}: representa el número de eventos diferentes que siguen al prefijo del elemento estudiado.
- O_{21}: representa el número de eventos iguales, de la misma longitud que siguen a un prefijo diferente.
- O_{22}: representa el número de eventos distintos, de la misma longitud que siguen a un prefijo diferente.

Los valores esperados se calculan mediante la siguiente fórmula:

$$Esperado(E_{ij}) = \frac{(Total\ Fila_i\ *\ Total\ Columna_j)}{Total}$$

Ecuación 1: Fórmula de cálculo del valor Chi-cuadrado (1).

Por último, el valor de Chi-cuadrado para cada elemento se obtiene utilizando la siguiente fórmula:

$$X^2 = \sum_{i=1}^{2} \sum_{j=1}^{2} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

Ecuación 2: Fórmula de cálculo del valor Chi-cuadrado (2).

, donde X^2 representa el valor Chi-cuadrado.

El último paso para la formación del modelo es la representación de los valores Chi-cuadrado de cada nodo en el trie. La siguiente ilustración muestra un trie con estos valores en cada nodo:

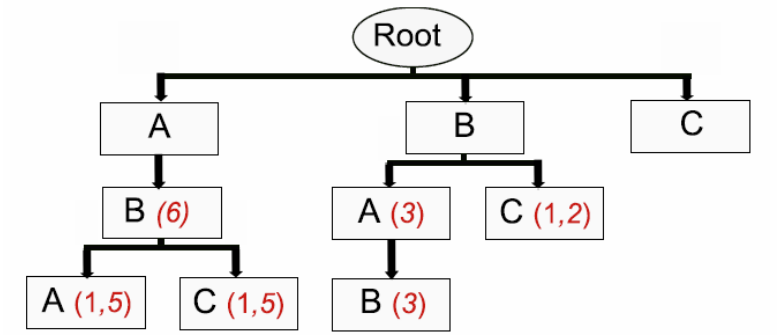

Ilustración 6: Trie con valores de Chi-cuadrado [62].

2.4.2. Modelado de Agentes Utilizando Secuenciación de Eventos: MAUSE

A diferencia del sistema M-COMP previamente descrito, el sistema MAUSE pretende crear un modelo de comportamiento a partir de una única secuencia de eventos ordenados. La Ilustración 7 muestra su arquitectura:

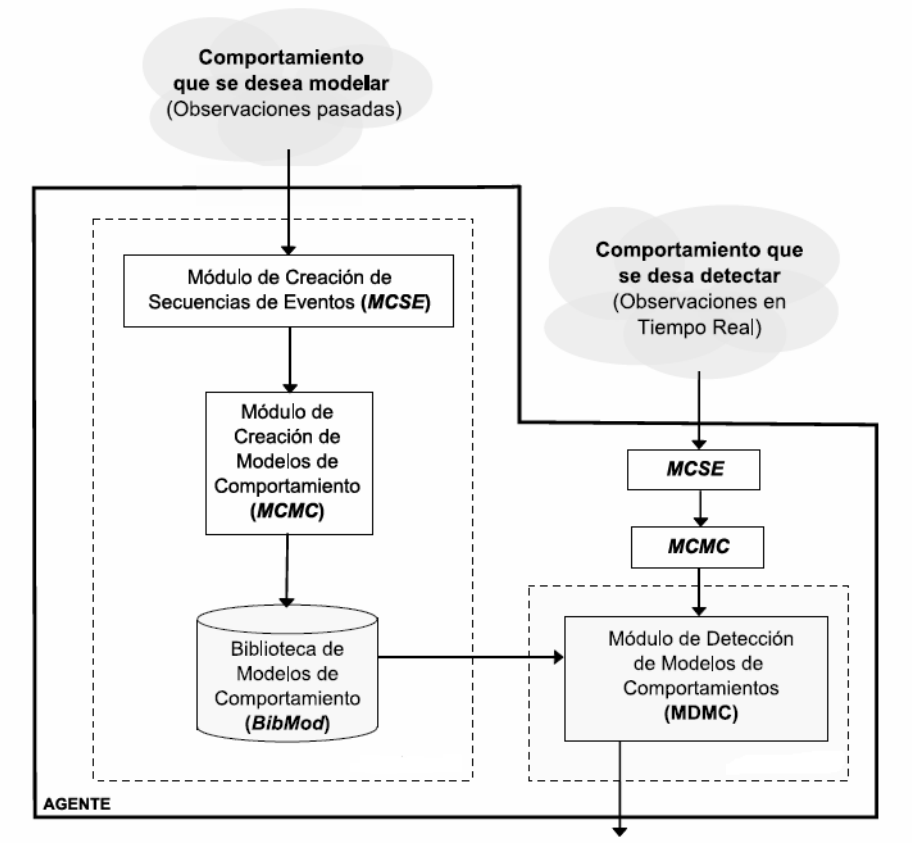

Ilustración 7: Sistema MAUSE [62].

A continuación se detallan los módulos relevantes a la realización de este proyecto. En este caso se trata de los mismos módulos explicados en la sección anterior, pero con pequeños cambios. Estos módulos son el Módulo de Creación de Secuencias de Eventos (MCSE) y el Módulo de Creación de Modelos de Comportamiento (MCMC).

2.4.2.1. Módulo de Creación de Secuencias de Eventos (MCSE)

Este módulo funciona exactamente igual que su homónimo definido en la *sección 2.4.1. Modelado de Agentes Mediante Comparación de Secuencias de Eventos: M-Comp*, por lo que su explicación se remite a la realizada en esa sección.

2.4.2.2. Módulo de Creación de Modelos de Comportamiento (MCMC)

Este módulo realiza las mismas funciones que su homónimo detallado en la sección *2.4.1. Modelado de Agentes Mediante Comparación de Secuencias de Eventos: M-Comp*, con la única diferencia de que el cálculo de la relevancia de un evento es diferente.

En este módulo, el cálculo de la relevancia de un evento se realiza mediante la frecuencia relativa de cada evento. Esta frecuencia se define como el número de veces que la subsecuencia se ha insertado en el trie, respecto al número total de secuencias de la misma longitud. A este tipo de frecuencia también se le denomina valor de soporte. En la sección *3.3.5.* se explica con detalle esta frecuencia.

El último paso que realiza este módulo es representar la relevancia de los eventos en el trie. La siguiente ilustración muestra un ejemplo de ello:

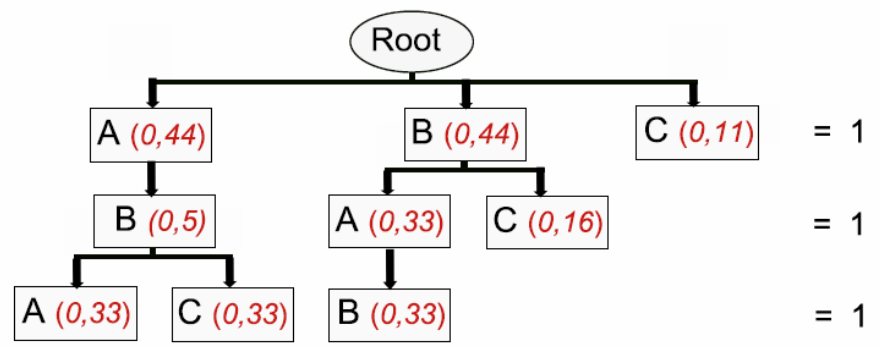

Ilustración 8: Trie con valores de soporte [62].

Capítulo 3: Descripción del sistema

3.1. Arquitectura del sistema

Para este proyecto se ha seguido el modelo arquitectónico definido como Modelo-Vista-Controlador.

Se trata de un patrón de diseño software en el que el sistema se divide y diseña en tres componentes: Modelo, Vista y Controlador.

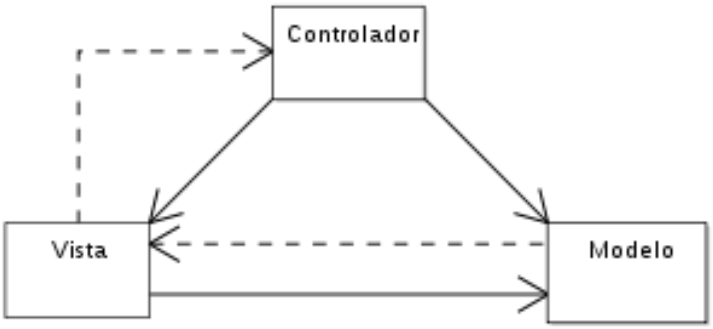

Ilustración 9: Arquitectura del sistema (Modelo-Vista-Controlador).

Modelo: contiene la representación de la información con la que el sistema opera, los requisitos definidos en las restricciones del sistema (la lógica de negocio) y todo el procesamiento de la información bajo las restricciones establecidas en la lógica de negocio.

Vista: se trata de la representación gráfica mediante una interfaz de usuario con la que un usuario es capaz de interactuar con el sistema. La Vista representa el modelo con el que el usuario es capaz de definir el formato de salida esperado del sistema (es capaz de configurar el sistema para que opere de la forma deseada).

Controlador: es el componente que conecta la Vista con el Modelo. Recibe los datos de la Vista e instancia el Modelo adecuado para que el sistema opere de la forma deseada.

El flujo de control entre los componentes es el siguiente:

1. El usuario interactúa con la interfaz, configurando todos los parámetros obligatorios y opcionales que ésta ofrece. Una vez que se pulsa el botón “Run”, se instancia el Controlador.
2. El Controlador recibe la configuración definida por la Vista e instancia el Modelo con todos los parámetros.
3. El Modelo instancia todas sus clases con los parámetros recibidos y el programa se ejecuta. Mientras éste se ejecuta, se imprimen en la interfaz los procesos intermedios que se van ejecutando, actualizando la Vista.
4. Cuando se finaliza todo el procesamiento por parte del Modelo, el Controlador deja de actualizar la Vista y ésta queda libre a disposición de futuras interacciones por parte del usuario.

3.2. Flujo de funcionamiento del sistema

Mediante el siguiente diagrama se representa el flujo que el Modelo del sistema sigue para su funcionamiento:

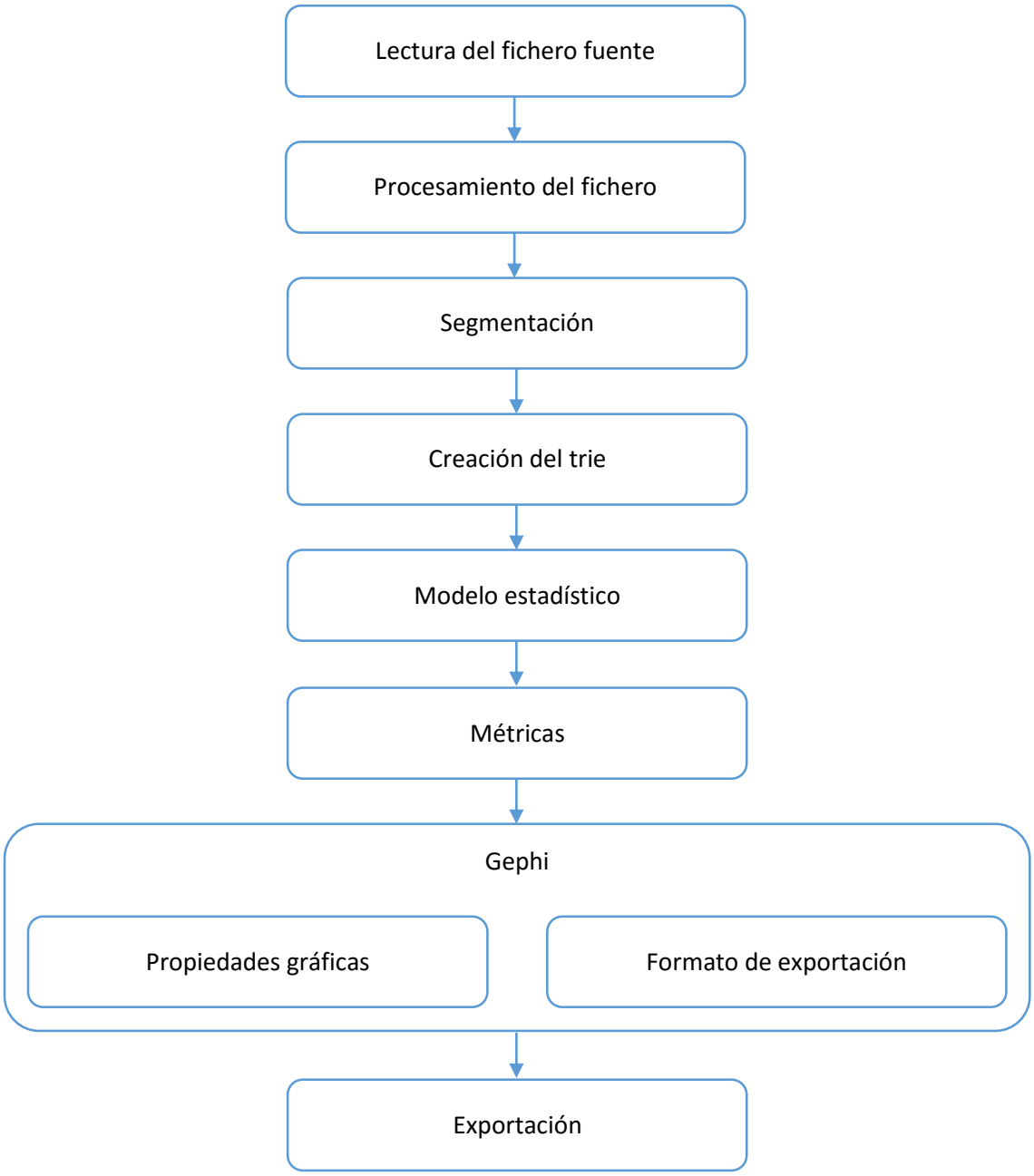

Ilustración 10: Flujo de funcionamiento del sistema.

A continuación se describen todos los procesos descritos en la ilustración anterior.

3.2.1. Lectura del fichero fuente

Este es el primer proceso del sistema en el que se lee y almacena la información del fichero fuente.

3.2.2. Procesamiento del fichero

En este proceso la información del fichero fuente se procesa y separa en acciones individuales, aprovechando el formato CSV del fichero fuente. Cada acción es almacenada en una posición individual de una estructura dinámica.

3.2.3. Segmentación

La parte de segmentación se compone de dos procesos para dividir el conjunto original de acciones en secuencias de acciones individuales. Estos procesos se han denominado segmentación por profundidad y segmentación por palabras clave.

3.2.3.1. Segmentación por profundidad

La segmentación por profundidad consiste en establecer el nivel máximo de profundidad que puede llegar a tener una secuencia de acciones individuales. Como profundidad se entiende el número de acciones individuales que forman una secuencia. Establecer una segmentación por profundidad de profundidad, por ejemplo tres, significa que todo el trie se formará con secuencias de acciones cuya longitud máxima será de tres acciones.

3.2.3.2. Segmentación por palabras clave

Este tipo de segmentación se realiza estableciendo una palabra inicial y una palabra final, que sirven como delimitadores para formar las secuencias de acciones del trie. Con este método de segmentación se crean secuencias de acciones que empiezan y acaban con las palabras clave establecidas. Se debe tener en cuenta que no todas las secuencias de acciones del trie empiezan y acaban con las palabras clave establecidas debido a la naturaleza de la formación de la estructura del trie. En el proceso de formación del trie se forman secuencias de acciones adicionales, que no necesariamente empiezan y acaban por las palabras clave establecidas.

3.2.4. Creación del trie

Para la creación del trie se ha seguido el método de formación explicado en la sección *2.4.1.2.*

3.2.5. Modelo estadístico

El enfoque estadístico desarrollado en este sistema sigue el implementado en la Tesis Doctoral *Modelado Automático del Comportamiento de Agentes Inteligentes* [62]. En concreto, se ha basado en el sistema MAUSE, detallado en la sección *2.4.2.*

Además de este modelo estadístico para representar la relevancia de un evento, se ha decidido implementar, lo que se denomina probabilidad de transición, detallada en la sección *3.2.5.2.*

3.2.5.1. Valores de soporte

El valor de soporte de una acción implica conocer el nivel de profundidad de cada nodo dentro del trie, o expresado de otra forma, la longitud de la secuencia de acciones que representa cada nodo desde el nodo raíz. La siguiente ilustración representa el trie dividido en niveles de profundidad:

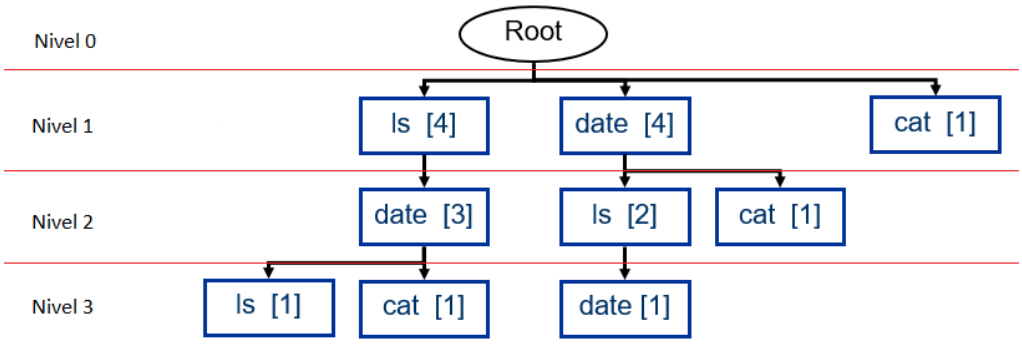

Ilustración 11: Niveles de profundidad del trie.

Por cada nodo se estudia su frecuencia de inserción respecto al total de frecuencias de inserción de todos los nodos del mismo nivel. La fórmula para calcular el valor de soporte de un nodo es:

$$soporte(x_i) = \frac{\text{frecuenciaInserción}(x_i)}{\sum_1^n \text{frecuenciaInserción}(x_{n,i})}$$

Ecuación 3: Fórmula de cálculo de los valores de soporte.

, donde *i* representa el nivel de profundidad y *n* cada nodo de esa profundidad.

Dicho esto, se ha creado la siguiente tabla que representan los cálculos necesarios para establecer el valor de soporte de cada nodo del trie:

Nodo	Nivel	Frecuencia de inserción del nodo	Total de frecuencia de inserción del nivel	Valor de soporte
ls	1	4	9	0,444
date	1	4	9	0,444
cat	1	1	9	0,111
date	2	3	6	0,500
ls	2	2	6	0,333
cat	2	1	6	0,167
ls	3	1	3	0,333
cat	3	1	3	0,333
date	3	1	3	0,333

Tabla 2: Tabla de cálculo de los valores de soporte.

La representación de los valores de soporte en el trie sería la siguiente:

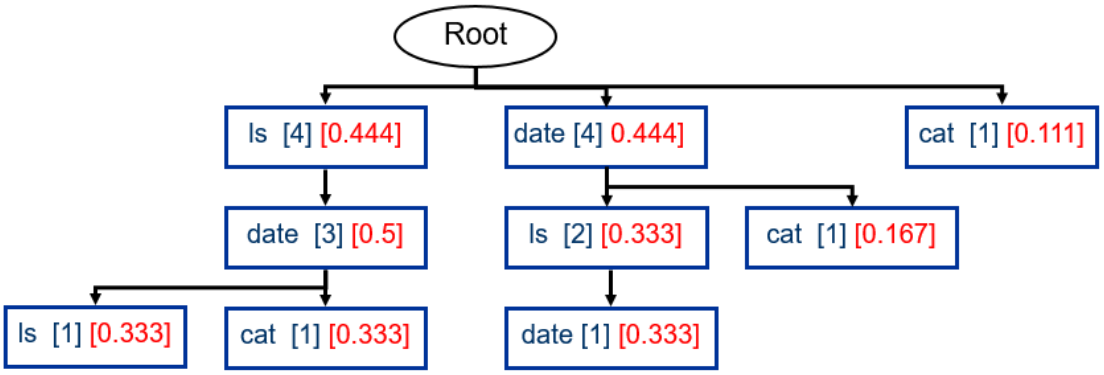

Ilustración 12: Representación del trie con valores de soporte.

3.2.5.2. Probabilidad de transición

La probabilidad de transición consiste en establecer la frecuencia relativa que tiene un nodo respecto al resto de nodos con el mismo nodo padre. Esta probabilidad viene representada como 1 entre el número de nodos con el mismo padre que el nodo estudiado. La siguiente tabla ilustra el cálculo de esta probabilidad:

Nodo	Nodo padre	Número de nodos hijos	Probabilidad de transición
ls	Root	3	0,333
date	Root	3	0,333
cat	Root	3	0,333
date	Root-ls	1	1,000
ls	Root-date	2	0,500
cat	Root-date	2	0,500
ls	Root-ls-date	2	0,500
cat	Root-ls-date	2	0,500
date	Root-date-ls	1	1,000

Tabla 3: Tabla de probabilidades de transición.

La representación de las probabilidades de transición en el trie quedaría como la siguiente:

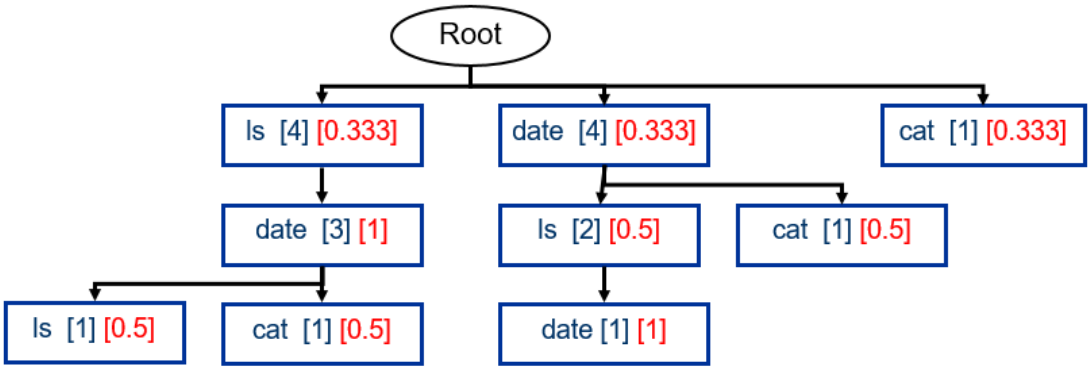

Ilustración 13: Representación del trie con probabilidades de transición.

3.2.6. Métricas

Este proceso se encarga de realizar el conjunto de métricas implementadas en este proyecto. Dada la naturaleza de investigación del proyecto se han desarrollado métricas enfocadas al análisis estadístico de las secuencias de acciones.

3.2.6.1. Frecuencia estadística

El objetivo de esta frecuencia es obtener todos los enlaces cuyos nodos destino cumplen cierta condición. Esta condición se denomina umbral. Todos los enlaces que igualen o superen este umbral son seleccionados y sus nodos destino son recogidos. Con frecuencia estadística se entiende la frecuencia seleccionada en el modelo estadístico: valores de soporte o probabilidades de transición.

Esta frecuencia está comprendida entre el intervalo representado por [0,1]. Además, se permite al usuario escoger un color para los enlaces que cumplen el umbral en la representación gráfica del trie.

3.2.6.2. Frecuencia de inserción

El objetivo de esta frecuencia es obtener todos los nodos que cumplen cierta condición. Esta condición puede ser “Todos”, obteniendo todas las frecuencias, o puede ser “Umbral”, donde se obtienen sólo aquellas frecuencias que igualan o superan el umbral indicado. Con frecuencia de inserción se entiende el número de veces que la acción se inserta en el trie en forma de nodo. La primera vez que se inserta la acción, su frecuencia de inserción es 1. A medida que se van insertando las mismas acciones en la misma posición del trie, esta frecuencia se incrementa.

Esta frecuencia está comprendida entre el intervalo de todos los números naturales, representado por [1, N]. Además, si se selecciona el criterio de umbral, se permite al usuario escoger un color para los nodos que cumplen el umbral en la representación gráfica del trie.

3.2.6.3. Eventos del dominio

El objetivo de esta métrica es obtener el conjunto de eventos individuales dentro del universo de eventos de un dominio. Con esta métrica se pueden obtener los eventos presentes y relevantes en un dominio, y descartar aquellos que no aparecen por esta métrica.

El funcionamiento de la métrica es el siguiente: cada evento nuevo que se lee desde el fichero fuente es almacenado, y los eventos ya almacenados no se vuelven a almacenar. Es por ello que, si se introduce un universo completo de eventos dentro de un dominio, todos los eventos que no aparezcan tras aplicar esta métrica, no forman parte del dominio o son irrelevantes para el dominio.

3.2.7. Gephi

En este proceso se crea la representación gráfica del trie mediante la API de Java de Gephi. La información de los nodos creados previamente es utilizada para construir las estructuras nodo y enlace propias de Gephi. Mediante la propia API, los nodos se pintan y distribuyen.

El subproceso denominado “Propiedades gráficas” se encarga de personalizar los nodos y enlaces (etiquetas, colores y curvatura) en función de las elecciones del usuario.

El subproceso “Formato de exportación” recibe la configuración de exportación que el usuario ha elegido y la pasa al último proceso “3.2.8. Exportación” para que el trie sea exportado.

En la formación gráfica del trie se ha definido que el nodo raíz o *Root* sea siempre de color rojo y tenga el doble de diámetro que el resto de nodos. De esta forma, el nodo raíz es siempre identificable en el trie.

El color del resto de los nodos y de los enlaces por defecto es negro. Estos colores se pueden definir mediantes los selectores de color de la interfaz o si se aplica alguna métrica con umbral, donde el color se aplica sobre aquellos elementos que tengan la misma frecuencia o mayor que el umbral indicado.

En la siguiente ilustración se muestra un ejemplo del pintado que realiza Gephi del trie:

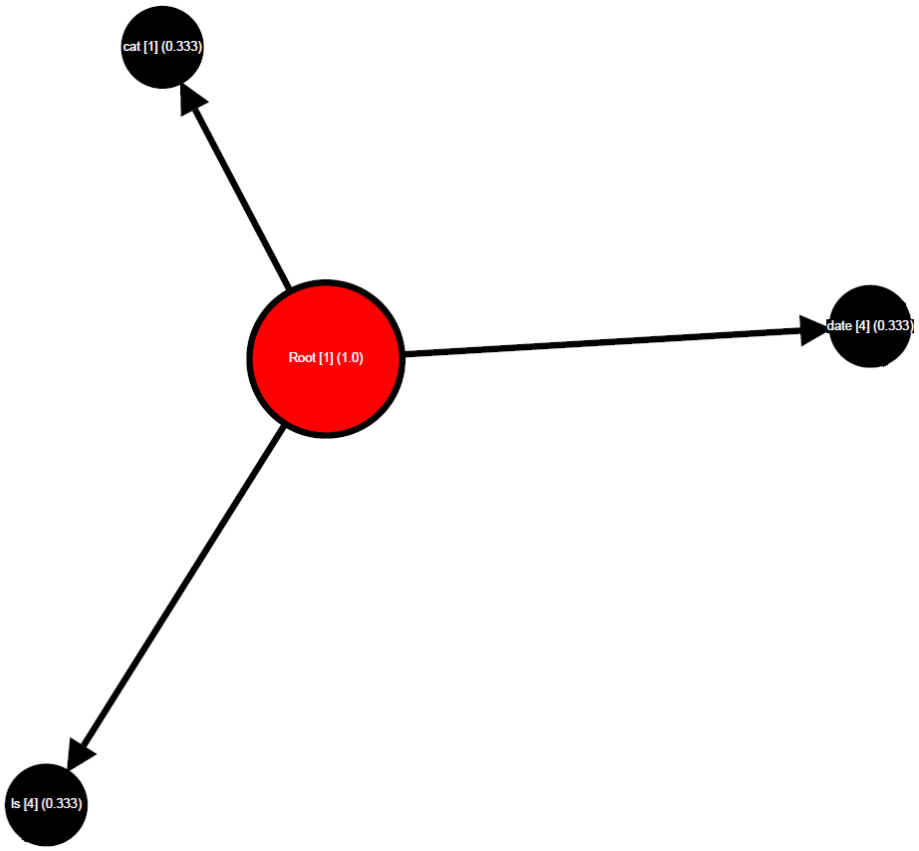

Ilustración 14: Representación gráfica del trie utilizando Gephi.

En la ilustración se han representado los siguientes elementos:

Elemento	Frecuencia de inserción	Frecuencia estadística
Root	1	1
ls	4	0,333
cat	1	0,333
date	4	0,333

Tabla 4: Elementos de la representación gráfica de ejemplo utilizando Gephi.

Este trie se ha formado con las acciones {*ls, cat, date*} y se ha especificado que el modelo estadístico sea Probabilidad de transición. Para el pintado se han activado las propiedades gráficas y se ha seleccionado el pintado de las etiquetas de los nodos. Estas etiquetas se forman con el siguiente formato:

Nombre de nodo [Frecuencia de inserción] (Frecuencia estadística)

Por último, los enlaces poseen pesos. Estos pesos se establecen como la frecuencia estadística del nodo destino. En el pintado del trie, los enlaces adquieren grosor en función del peso que tienen. A mayor peso, mayor grosor.

3.2.8. Exportación

Este es el último proceso que se realiza, obteniendo la representación gráfica y el método de exportación del proceso "3.2.7. Gephi". El proceso se encarga de exportar el trie y sus pertinentes ficheros en el directorio donde se ubica la herramienta. Los formatos de exportación disponibles son *PDF* o *gexf* (formato propio de Gephi). El resto de ficheros (CSV con todos los nodos del trie y ficheros de métricas) que se forman son ficheros de texto en formato de texto plano *txt*.

3.3. Restricciones del sistema

En esta sección se detallan las restricciones hardware y software por las que el sistema se rige.

3.3.1. Restricciones hardware

Debido a que el sistema precisa únicamente de un ordenador, no se requieren restricciones hardware de otros dispositivos físicos. A continuación se listan las restricciones hardware:

- El ordenador en el que se ejecuta la herramienta debe poseer monitor, teclado y ratón mediante los cuales el usuario será capaz de interactuar con la interfaz gráfica.
- Se debe disponer de espacio de memoria suficiente como para albergar los ficheros de texto resultantes de la ejecución del software.
- Se debe disponer de al menos 2GB de memoria RAM para que el software pueda operar con grandes cantidades de datos en tiempos relativamente cortos.

3.3.2. Restricciones software

Cuando se ejecute el software, se deben considerar las siguientes restricciones:

- El sistema operativo del ordenador debe ser capaz de ejecutar programas Java.
- Se debe disponer de las versiones de Java 1.7 y 1.8, ya que el programa se ejecuta bajo Java 8 y Gephi (en caso de usar su aplicación de escritorio) se ejecuta bajo Java 7.
- La colocación de los nodos depende exclusivamente del algoritmo de distribución que se aplica. Este algoritmo aplica un determinado número de iteraciones sobre los nodos con el objetivo de redistribuirlos por el espacio. Cada vez que el algoritmo se aplica, la posición absoluta de los nodos cambia, por lo que ejecutando el programa dos veces sobre el mismo conjunto de datos, no necesariamente se representa la misma distribución de nodos. Además, en ocasiones los enlaces se cruzan con los nodos, por lo que la representación gráfica puede llegar a ser confusa.

- Las etiquetas de los enlaces no se pueden mover respecto al enlace al que acompañan, y en caso de activarlas éstas se posicionan en el centro del enlace (tanto de longitud, como de altura).
- Cuando se especifican enlaces curvados, Gephi pierde la direccionalidad en la representación gráfica. El grafo resultante tiene los enlaces curvos, pero no posee las flechas de los enlaces. Esto no significa que el grafo deja de ser dirigido, sino que la API de Gephi posee esta limitación en cuanto a la exportación gráfica en PDF del trie. Si el grafo se abre utilizando la aplicación de escritorio de Gephi, los enlaces poseen flechas, por lo que la direccionalidad se conserva.

3.4. Casos de uso

En esta sección se definen los casos de uso del sistema implementado. El siguiente diagrama muestra el diagrama completo de los casos de usos:

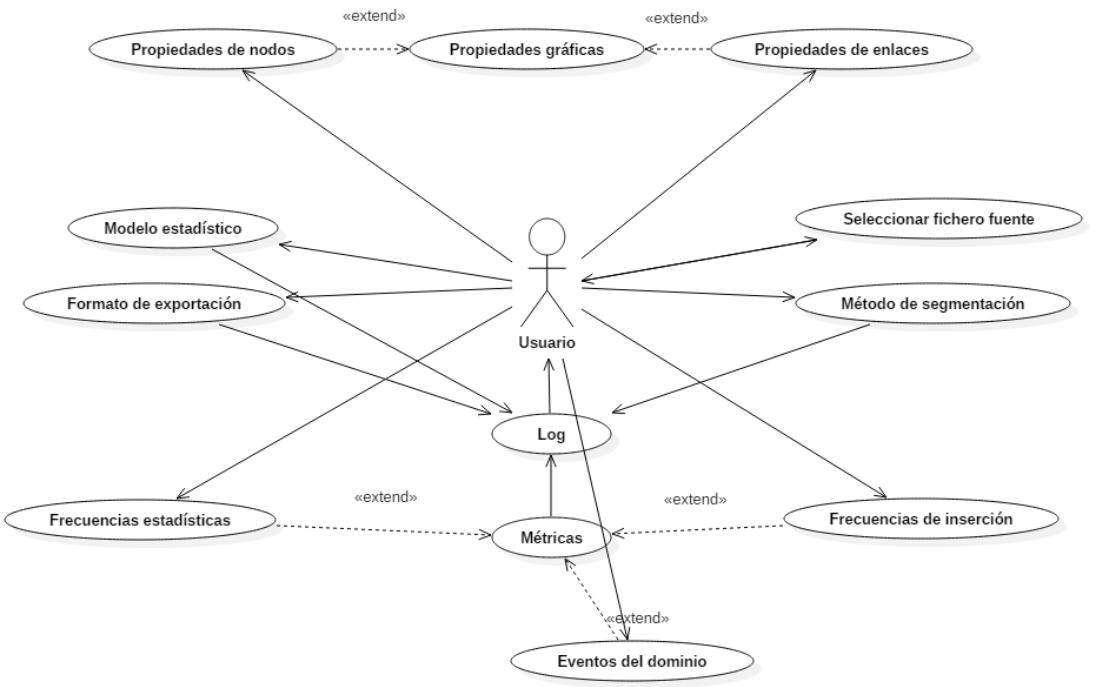

Ilustración 15: Diagrama completo de casos de uso.

A continuación se describe la estructura que define cada caso de uso:

- **Código**: se trata de un código único por el que un caso de uso puede ser identificado rápidamente. Este código se compone de las iniciales CU (de Caso de Uso) y de un identificador numérico consecutivo expresado con tres dígitos numéricos. Un ejemplo de formato de código es el siguiente: CU-001, donde se expresa que se trata de un caso de uso y tiene como orden el primer caso de uso.
- **Nombre**: el nombre permite realizar una breve descripción del caso de uso estudiado.
- **Actor**: define el rol que realiza este caso de uso. En este proyecto existe únicamente el rol de Usuario. Cualquier usuario que interaccione con el programa activará el funcionamiento de todo el sistema.

- **Descripción**: este campo contiene toda la descripción del caso de uso.
- **Precondiciones**: en este campo se describen las condiciones previas del sistema que se deben cumplir para que se dé el caso de uso.
- **Postcondiciones**: en este campo se describen los efectos que se producen cuando se presenta el caso de uso.

Mediante tablas que se componen de estos campos se describen todos los casos de uso del sistema:

Código	CU-001
Nombre	Seleccionar fichero fuente
Actor	Usuario
Descripción	El usuario selecciona un fichero fuente de acciones
Precondiciones	Ninguna
Postcondiciones	- El fichero fuente se ha seleccionado correctamente y su ruta es impresa en el campo de texto de la ruta. -Se guarda la ruta del fichero seleccionado.

Tabla 5: Caso de uso CU-001.

Código	CU-002
Nombre	Método de segmentación
Actor	Usuario
Descripción	El usuario elige un método de segmentación.
Precondiciones	Ninguna
Postcondiciones	- Se marca el método de segmentación escogido junto con sus parámetros. - Se guarda la selección del usuario. - Cuando se ejecuta el programa se muestra el proceso de segmentación.

Tabla 6: Caso de uso CU-002.

Código	CU-003
Nombre	Modelo estadístico
Actor	Usuario
Descripción	El usuario elige un modelo estadístico.
Precondiciones	Ninguna
Postcondiciones	- Se marca el modelo estadístico escogido - Se guarda la selección del usuario. - Cuando se ejecuta el programa se crean los ficheros necesarios.

Tabla 7: Caso de uso CU-003.

Código	CU-004
Nombre	Formato de exportación
Actor	Usuario
Descripción	El usuario elige un formato de exportación o ambos formatos de exportación.
Precondiciones	Ninguna
Postcondiciones	- Se marca / n el / los formato / s de exportación escogidos. - Se guarda la selección del usuario.

Tabla 8: Caso de uso CU-004.

Código	CU-005
Nombre	Propiedades de nodos
Actor	Usuario
Descripción	El usuario activa / desactiva las propiedades gráficas respecto al pintado de los nodos.
Precondiciones	Se deben activar las propiedades gráficas
Postcondiciones	- Se marcan o desmarcan las propiedades gráficas de los nodos. - Se guarda la selección del usuario.

Tabla 9: Caso de uso CU-005.

Código	CU-006
Nombre	Propiedades de enlaces
Actor	Usuario
Descripción	El usuario activa / desactiva las propiedades gráficas respecto al pintado de los enlaces.
Precondiciones	Se deben activar las propiedades gráficas
Postcondiciones	- Se marcan o desmarcan las propiedades gráficas de los enlaces. - Se guarda la selección del usuario.

Tabla 10: Caso de uso CU-006.

Código	CU-007
Nombre	Propiedades gráficas
Actor	Usuario
Descripción	El usuario activa / desactiva las propiedades gráficas respecto al pintado de los nodos y enlaces.
Precondiciones	Ninguna
Postcondiciones	Se activan / desactivan las propiedades gráficas de los nodos y enlaces.

Tabla 11: Caso de uso CU-007.

Código	CU-008
Nombre	Frecuencias estadísticas
Actor	Usuario
Descripción	El usuario define los parámetros de la métrica "Frecuencias estadísticas".
Precondiciones	El usuario activa / desactiva la métrica.
Postcondiciones	Se guarda la selección del usuario.

Tabla 12: Caso de uso CU-008.

Código	CU-009
Nombre	Frecuencias de inserción
Actor	Usuario
Descripción	El usuario define los parámetros de la métrica "Frecuencias de inserción".
Precondiciones	El usuario activa / desactiva la métrica
Postcondiciones	Se guarda la selección del usuario.

Tabla 13: Caso de uso CU-009.

Código	CU-010
Nombre	Acciones individuales
Actor	Usuario
Descripción	El usuario activa / desactiva la métrica "Acciones individuales".
Precondiciones	Ninguna
Postcondiciones	Se guarda la selección del usuario.

Tabla 14: Caso de uso CU-010.

Código	CU-011
Nombre	Métricas
Actor	Usuario
Descripción	Caso de uso que define el comportamiento cuando el usuario ejecuta el programa con alguna métrica seleccionada.
Precondiciones	El usuario ejecuta el programa.
Postcondiciones	- Las métricas seleccionadas se imprimen en el área de texto del Log. - Se crean los ficheros de texto resultantes en el directorio de ejecución.

Tabla 15: Caso de uso CU-011.

Código	CU-012
Nombre	Log
Actor	Usuario
Descripción	Caso de uso que define el comportamiento cuando el usuario ejecuta el programa.
Precondiciones	El usuario ejecuta el programa.
Postcondiciones	- Se recogen todas las impresiones de los procesos intermedios y se muestran al usuario en el área de texto del campo Log.

Tabla 16: Caso de uso CU-012.

3.5. Requisitos del sistema

En esta sección se van a detallar todos los requisitos que delimitan el funcionamiento del sistema implementado. Estos requisitos se van a separar en requisitos funcionales y requisitos no funcionales.

Los requisitos funcionales son los que determinan el comportamiento del sistema. Engloban el modelo de interacción con el usuario y el funcionamiento del sistema. Son los que definen qué debe hacer el sistema.

Por otro lado, los requisitos no funcionales son los que definen la estructura del sistema. Estos requisitos controlan cualidades como la organización, el rendimiento o la interfaz de un sistema. Son los requisitos que definen cómo funciona el sistema.

Todos los requisitos se van a presentar en formato de tablas individuales donde se muestran las siguientes características:

- **Código**: se trata de un código único por el que un requisito puede ser identificado rápidamente. Este código se compone del tipo de requisito que se está tratando y de un identificador numérico consecutivo expresado con tres dígitos numéricos. Un ejemplo de formato de código es el siguiente: RF-001, donde se expresa que se trata de un requisito funcional y tiene como orden el primer requisito.

- **Caso de uso**: presente únicamente en los requisitos funcionales, permite referenciar al caso de uso del que se ha extraído el requisito funcional, especificando su código.
- **Nombre**: el nombre permite realizar una breve descripción del requisito estudiado.
- **Descripción**: este campo contiene toda la descripción del requisito. Aquí se detallan los comportamientos esperados y los flujos de acciones necesarios para que se realicen las acciones descritas por el requisito.
- **Prioridad**: la prioridad de un requisito indica su nivel de importancia en la implementación del proyecto. Existen tres niveles de prioridad:
    - Alta: indica que el requisito es esencial para el proyecto, y que su implementación debe ser llevada a cabo para cumplir con los objetivos del proyecto.
    - Media: indica que el requisito no es esencial, pero es deseable que fuera implementado, aunque sea en la fase final del proyecto. Los requisitos con esta prioridad no son determinantes para cumplir con los objetivos del proyecto.
    - Baja: indica que el requisito es opcional. Su implementación se realizará en la fase de refinamiento final del proyecto, ya que no influye en los objetivos del proyecto en ninguna medida.
- **Verificabilidad**: los requisitos pueden ser más o menos complicados de verificar y validar por lo que es necesario llevar un control de cuál es el grado de dificultad en verificar que un requisito se ha implementado correctamente. Se han definido tres grados de verificabilidad:
    - Alta: indica que el requisito es fácil y rápidamente verificable.
    - Media: indica que el requisito es algo lento y difícil de verificar.
    - Baja: indica que el requisito es muy tedioso y complejo de verificar.

A continuación se listan los requisitos del sistema.

3.5.1. Requisitos funcionales

Código	RF-001	Caso de uso	CU-001, CU-002, CU-003, CU-004
Nombre	Ejecución completa.		
Descripción	Para que el programa se ejecute es necesario que el usuario seleccione un fichero válido (acorde a las restricciones de formato), un método de segmentación, un modelo estadístico y al menos un formato de exportación. El resto de parámetros de configuración son opcionales. Una vez se hayan elegido todos los parámetros es necesario pulsar el botón “Run” para ejecutar el programa.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 17: Requisito funcional RF-001.

Código	RF-002	Caso de uso	CU-001, CU-002, CU-003, CU-004
Nombre	Ejecución incompleta.		
Descripción	En caso de que no se seleccionen todos los parámetros obligatorios, se mostrarán los errores pertinentes.		
Prioridad	Media	Verificabilidad	Alta

Tabla 18: Requisito funcional RF-002.

Código	RF-003	Caso de uso	CU-001
Nombre	Selección de fichero.		
Descripción	Cuando el botón de seleccionar fichero es pulsado, se debe abrir el selector de ficheros del sistema. Si se selecciona un fichero, su ruta absoluta debe ser mostrada en el cuadro de texto situado al lado del botón de seleccionar fichero.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 19: Requisito funcional RF-003.

Código	RF-004	Caso de uso	CU-002
Nombre	Método de segmentación: activación de campos.		
Descripción	En función del método de segmentación seleccionado, se debe / n activar el / los campo / s de texto correspondiente / s.		
Prioridad	Media	Verificabilidad	Alta

Tabla 20: Requisito funcional RF-004.

Código	RF-005	Caso de uso	CU-002
Nombre	Método de segmentación: cambio de método.		
Descripción	Tras elegir un método de segmentación y cambiar a otro, al volver al primero se reestablecerán los valores previamente establecidos (profundidad o palabras clave).		
Prioridad	Baja	Verificabilidad	Alta

Tabla 21: Requisito funcional RF-005.

Código	RF-006	Caso de uso	CU-002
Nombre	Método de segmentación: profundidad.		
Descripción	Si se selecciona el método de segmentación por profundidad, se realizará el trie siguiendo el método de segmentación por profundidad con la profundidad indicada.		
Prioridad	Alta	Verificabilidad	Media

Tabla 22: Requisito funcional RF-006.

Código	RF-007	Caso de uso	CU-002
Nombre	Método de segmentación: palabras clave.		
Descripción	Si se selecciona el método de segmentación por palabras clave, se realizará el trie siguiendo el método de segmentación por palabras clave con las palabras clave indicadas.		
Prioridad	Alta	Verificabilidad	Media

Tabla 23: Requisito funcional RF-007.

Código	RF-008	Caso de uso	CU-003
Nombre	Modelo estadístico: valores de soporte.		
Descripción	Si se selecciona el método estadístico de valores de soporte, el trie se realizará siguiendo el modelo estadístico de valores de soporte. Además, se debe crear un fichero con extensión *txt* que contenga todos los nodos (a excepción de *Root*), ordenados según su valor de soporte, en orden descendente. Este fichero se debe denominar: SupportValues_aaaa-mm-dd_hh-mm-ss.txt.		
Prioridad	Alta	Verificabilidad	Media

Tabla 24: Requisito funcional RF-008.

Código	RF-009	Caso de uso	CU-003
Nombre	Modelo estadístico: probabilidades de transición.		
Descripción	Si se selecciona el método estadístico de probabilidades de transición, el trie se realizará siguiendo el modelo estadístico de probabilidades de transición. Además, se debe crear un fichero con extensión *txt* que contenga todos los nodos (a excepción de *Root*), ordenados según su probabilidad de transición, en orden descendente. Este fichero se debe denominar: TransitionProbabilities_aaaa-mm-dd_hh-mm-ss.txt.		
Prioridad	Alta	Verificabilidad	Media

Tabla 25: Requisito funcional RF-009.

Código	RF-010	Caso de uso	CU-004
Nombre	Formato de exportación: PDF.		
Descripción	Si se selecciona el formato de exportación como PDF, el trie debe exportarse en un fichero con extensión *PDF* con el siguiente formato: aaaa-mm-dd_hh-mm-ss.pdf		
Prioridad	Alta	Verificabilidad	Alta

Tabla 26: Requisito funcional RF-010.

Código	RF-011	Caso de uso	CU-004
Nombre	Formato de exportación: Gephi.		
Descripción	Si se selecciona el formato de exportación como Gephi, el trie debe exportarse en un fichero con extensión *gexf* con el siguiente formato: aaaa-mm-dd_hh-mm-ss.gexf		
Prioridad	Alta	Verificabilidad	Alta

Tabla 27: Requisito funcional RF-011.

Código	RF-012	Caso de uso	CU-001, CU-002, CU-003, CU-004
Nombre	Exportación CSV.		
Descripción	Sea cual sea el formato de exportación seleccionado, se debe exportar además en formato CSV la siguiente información del trie generado: - El número total de nodos y enlaces que forman el trie. - Todos los nodos del trie, representando un nodo por línea con toda su información: identificador, frecuencia de inserción, ruta y, en función del modelo estadístico seleccionado, probabilidades de transición o valores de soporte. Este fichero debe tener extensión *txt* con el siguiente formato: TrieData_aaaa-mm-dd_hh-mm-ss.txt		
Prioridad	Alta	Verificabilidad	Media

Tabla 28: Requisito funcional RF-012.

Código	RF-013	Caso de uso	CU-007
Nombre	Propiedades gráficas y métricas: estado por defecto.		
Descripción	Por defecto, tanto las propiedades gráficas como todas las métricas deben estar desactivadas.		
Prioridad	Media	Verificabilidad	Alta

Tabla 29: Requisito funcional RF-013.

Código	RF-014	Caso de uso	CU-007
Nombre	Propiedades gráficas: activación.		
Descripción	Cuando se selecciona el radio botón de propiedades personalizadas se deben activar todas las opciones correspondientes al pintado de nodos y enlaces (a excepción de los botones de color, tanto de nodos como de enlaces).		
Prioridad	Alta	Verificabilidad	Alta

Tabla 30: Requisito funcional RF-014.

Código	RF-015	Caso de uso	CU-007
Nombre	Propiedades gráficas: desactivación.		
Descripción	Cuando se deselecciona el radio botón de propiedades personalizadas se deben desactivar todas las opciones correspondientes al pintado de nodos y enlaces (incluidos los botones de color), almacenando los valores previamente establecidos, pero estableciendo los valores por defecto en todas las propiedades.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 31: Requisito funcional RF-015.

Código	RF-016	Caso de uso	CU-007, CU-011
Nombre	Activación botón de color: propiedades gráficas y métricas.		
Descripción	Cuando se selecciona la opción de color (propiedades gráficas) o la opción de umbral (métricas), el botón de seleccionar color se debe activar para ser pulsable.		
Prioridad	Media	Verificabilidad	Alta

Tabla 32: Requisito funcional RF-016.

Código	RF-017	Caso de uso	CU-007, CU-011
Nombre	Selector de colores: propiedades gráficas y métricas.		
Descripción	Con el botón de coloración activo, si éste se pulsa, se debe abrir un selector de colores en el que se puede seleccionar un color. Si se pulsa "Aceptar" en ese selector, el selector se debe cerrar y en el campo de previsualización de color se debe mostrar el color seleccionado. Si se selecciona "Cancelar" el selector de colores se debe cerrar sin haber cambiado el color que había previamente seleccionado.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 33: Requisito funcional RF-017.

Código	RF-018	Caso de uso	CU-007, CU-011
Nombre	Color nodos y enlaces por defecto tras desactivación de propiedades gráficas o métricas.		
Descripción	Si el campo de color de las propiedades gráficas o de las métricas está desactivado, sea por deshabilitar esa opción en concreto o por deshabilitar todas las propiedades, los colores de los nodos y enlaces volverán a reestablecerse a negro.		
Prioridad	Media	Verificabilidad	Media

Tabla 34: Requisito funcional RF-018.

Código	RF-019	Caso de uso	CU-005
Nombre	Propiedades gráficas: etiquetas de nodos.		
Descripción	Cuando se selecciona la opción de etiquetas de los nodos, en la representación gráfica del trie se deben mostrar las etiquetas de los nodos, en color blanco y dentro de los nodos, con la siguiente información: Identificador [frecuencia de inserción] (frecuencia estadística)		
Prioridad	Alta	Verificabilidad	Media

Tabla 35: Requisito funcional RF-019.

Código	RF-020	Caso de uso	CU-005
Nombre	Propiedades gráficas: etiquetas de nodos por defecto.		
Descripción	Las etiquetas de los nodos por defecto deben estar desactivadas y no se deberían de ver en la representación gráfica del trie.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 36: Requisito funcional RF-020.

Código	RF-021	Caso de uso	CU-005
Nombre	Propiedades gráficas: color de nodos.		
Descripción	Tras haber seleccionado el color de los nodos desde el selector de colores, el trie resultante debe tener los nodos del color escogido.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 37: Requisito funcional RF-021.

Código	RF-022	Caso de uso	CU-005
Nombre	Propiedades gráficas: color de nodos por defecto.		
Descripción	El color de los nodos por defecto debe ser negro.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 38: Requisito funcional RF-022.

Código	RF-023	Caso de uso	CU-006
Nombre	Propiedades gráficas: etiquetas de enlaces.		
Descripción	Cuando se selecciona la opción de etiquetas en los enlaces, en la representación gráfica del trie se debe mostrar, en el centro de los enlaces y en color negro, la frecuencia estadística seleccionada del nodo destino del enlace.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 39: Requisito funcional RF-023.

Código	RF-024	Caso de uso	CU-006
Nombre	Propiedades gráficas: etiquetas de enlaces por defecto.		
Descripción	Las etiquetas de los enlaces por defecto deben estar desactivadas y no se deberían de ver en la representación gráfica del trie.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 40: Requisito funcional RF-024.

Código	RF-025	Caso de uso	CU-006
Nombre	Propiedades gráficas: color de enlaces.		
Descripción	Tras haber seleccionado el color de los enlaces desde el selector de colores, el trie resultante debe tener los enlaces del color escogido.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 41: Requisito funcional RF-025.

Código	RF-026	Caso de uso	CU-006
Nombre	Propiedades gráficas: color de enlaces por defecto.		
Descripción	El color de los enlaces por defecto debe ser negro.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 42: Requisito funcional RF-026.

Código	RF-027	Caso de uso	CU-006
Nombre	Propiedades gráficas: curvatura de enlaces.		
Descripción	Si se selecciona esta opción, en la representación gráfica del trie, los enlaces se deben mostrar curvados.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 43: Requisito funcional RF-027.

Código	RF-028	Caso de uso	CU-006
Nombre	Propiedades gráficas: curvatura de enlaces por defecto.		
Descripción	La curvatura de los enlaces por defecto debe estar desactivada.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 44: Requisito funcional RF-028.

Código	RF-029	Caso de uso	CU-011
Nombre	Métricas: activación.		
Descripción	Si se selecciona el radio botón "*Enabled*" de las métricas, tanto de frecuencias estadísticas, como de frecuencias de inserción, se deben activara las opciones y campos disponibles, a excepción de los botones de color.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 45: Requisito funcional RF-029.

Código	RF-030	Caso de uso	CU-011
Nombre	Métricas: desactivación.		
Descripción	Cuando se deselecciona el radio botón de las métricas "*Enabled*" (sea de las frecuencias estadísticas o frecuencias de inserción), se deben desactivar todas las opciones y campos de las métricas correspondientes. Además, se deben almacenar los valores previamente establecidos, pero no se deben realizar las métricas deseleccionadas.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 46: Requisito funcional RF-030.

Código	RF-031	Caso de uso	CU-011
Nombre	Métricas: color por defecto.		
Descripción	El color por defecto de las métricas debe ser negro.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 47: Requisito funcional RF-031.

Código	RF-032	Caso de uso	CU-008
Nombre	Métricas: frecuencias estadísticas con umbral mínimo.		
Descripción	Cuando se selecciona la opción frecuencias estadísticas con umbral, se deben mostrar todas las frecuencias estadísticas del trie que igualan o superan este umbral. Si además se ha seleccionado un color, en la representación gráfica del trie se deben visualizar aquellos enlaces que igualan o superan el umbral indicado.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 48: Requisito funcional RF-032.

Código	RF-033	Caso de uso	CU-009
Nombre	Métricas: todas las frecuencias de inserción.		
Descripción	Cuando se selecciona la opción de todas las frecuencias de inserción se deben mostrar todas las frecuencias de inserción del trie.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 49: Requisito funcional RF-033.

Código	RF-034	Caso de uso	CU-009
Nombre	Métricas: frecuencias de inserción con umbral mínimo.		
Descripción	Cuando se selecciona la opción frecuencias de inserción con umbral, se deben mostrar todas las frecuencias de inserción del trie que igualan o superan este umbral. Si además se ha seleccionado un color, en la representación gráfica del trie se deben visualizar aquellos nodos que igualan o superan el umbral indicado.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 50: Requisito funcional RF-034.

Código	RF-035	Caso de uso	CU-011
Nombre	Métricas: activación acciones individuales.		
Descripción	Cuando se selecciona la opción de acciones individuales, se deben obtener y mostrar las acciones individuales de todo el conjunto de acciones originales.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 51: Requisito funcional RF-035.

Código	RF-036	Caso de uso	CU-011
Nombre	Métricas: desactivación acciones individuales.		
Descripción	Cuando se deselecciona la opción de acciones individuales, la métrica no se debe realizar.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 52: Requisito funcional RF-036.

Código	RF-037	Caso de uso	CU-011
Nombre	Métricas: exportación.		
Descripción	Todas las métricas seleccionadas deben generar un fichero en formato *txt* que contiene los datos generados por la métrica. Este fichero se debe nombrar con el siguiente formato: NombreMetrica_aaaa-mm-dd_hh-mm-ss.txt, donde NombreMetrica varía en función de la métrica seleccionada: - Todas las frecuencias de inserción: InsertionFrequenciesAll - Las frecuencias de inserción con umbral: InsertionFrequenciesWithThreshold - Las frecuencias estadísticas con umbral: StatisticalFrequenciesWithThreshold - Eventos del dominio: DomainEvents		
Prioridad	Alta	Verificabilidad	Alta

Tabla 53: Requisito funcional RF-037.

3.5.2. Requisitos no funcionales

Código	RNF-001		
Nombre	Formato de fichero de entrada.		
Descripción	El fichero de entrada de datos debe tener el formato CSV, donde las acciones individuales sean representadas en la primera línea de un fichero con formato ".txt", separadas por el símbolo de coma ",". Se debe tener una única línea con todas las acciones.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 54: Requisito no funcional RNF-001.

Código	RNF-002		
Nombre	Formato de acciones: separación.		
Descripción	El formato de las acciones individuales no debe contener el símbolo guion "-", ya que éste es procesado internamente como separador de las acciones individuales.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 55: Requisito no funcional RNF-002.

Código	RNF-003		
Nombre	Formato de acciones: longitud.		
Descripción	Las acciones deben contener al menos un carácter de longitud y que no sea blanco " ".		
Prioridad	Alta	Verificabilidad	Alta

Tabla 56: Requisito no funcional RNF-003.

Código	RNF-004		
Nombre	Dimensiones de interfaz de usuario.		
Descripción	La dimensión de la interfaz gráfica debe ser de 1024x670 píxeles. No debe ser posible redimensionar la interfaz.		
Prioridad	Media	Verificabilidad	Alta

Tabla 57: Requisito no funcional RNF-004.

Código	RNF-005		
Nombre	Posicionamiento de interfaz de usuario.		
Descripción	La interfaz gráfica se debe centrar automáticamente en el centro de la pantalla.		
Prioridad	Media	Verificabilidad	Alta

Tabla 58: Requisito no funcional RNF-005.

Código	RNF-006		
Nombre	Pantalla Log.		
Descripción	Debe existir un área de texto no editable que muestre todos los errores y procesos intermedios durante la ejecución del programa.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 59: Requisito no funcional RNF-006.

Código	RNF-007		
Nombre	Pantalla Log: ajuste de línea.		
Descripción	La pantalla de Log debe tener activado el ajuste de línea para que no exista un scroll horizontal.		
Prioridad	Media	Verificabilidad	Alta

Tabla 60: Requisito no funcional RNF-007.

Código	RNF-008		
Nombre	Pantalla Log: scroll vertical.		
Descripción	La pantalla de Log debe tener scroll vertical para poder visualizar todo el contenido de la pantalla.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 61: Requisito no funcional RNF-008.

Código	RNF-009		
Nombre	Pantalla Log: scroll automático.		
Descripción	La pantalla de Log debe tener un scroll automático durante la ejecución donde siempre se muestre la última línea.		
Prioridad	Media	Verificabilidad	Alta

Tabla 62: Requisito no funcional RNF-009.

Código	RNF-010		
Nombre	Selector de ficheros.		
Descripción	Debe existir un botón que al ser pulsado despliegue el selector de ficheros del sistema. Los ficheros mostrados deben ser filtrados y se deben mostrar únicamente los que tienen como extensión ".txt".		
Prioridad	Media	Verificabilidad	Alta

Tabla 63: Requisito no funcional RNF-010.

Código	RNF-011		
Nombre	Botón Run.		
Descripción	Debe existir un botón que al ser pulsado ejecute todo el proceso con la configuración establecida por el usuario.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 64: Requisito no funcional RNF-011.

Código	RNF-012		
Nombre	Ruta de fichero seleccionado.		
Descripción	Debe existir un campo de texto que contenga la ruta absoluta del fichero seleccionado desde el selector de ficheros. Este campo debe ser situado al lado del botón del selector de ficheros.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 65: Requisito no funcional RNF-012.

Código	RNF-013		
Nombre	Método de segmentación: exclusividad.		
Descripción	Se debe poder elegir únicamente un método de segmentación de entre los dos disponibles: segmentación por profundidad o segmentación por palabras clave.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 66: Requisito no funcional RNF-013.

Código	RNF-014		
Nombre	Método de segmentación: formato profundidad.		
Descripción	El campo de texto para la profundidad debe aceptar únicamente dígitos numéricos que forman números enteros y positivos. El resto de caracteres deben ser identificados como error de formato.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 67: Requisito no funcional RNF-014.

Código	RNF-015		
Nombre	Método de segmentación: formato palabras clave.		
Descripción	Los campos de texto para las palabras clave deben aceptar todo tipo de caracteres.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 68: Requisito no funcional RNF-015.

Código	RNF-016		
Nombre	Método de segmentación: profundidad demasiado alta.		
Descripción	Si no existe ninguna secuencia de acciones con la profundidad indicada, se debe mostrar un error y todo el proceso de ejecución se debe detener.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 69: Requisito no funcional RNF-016.

Código	RNF-017		
Nombre	Método de segmentación: palabra clave inexistente.		
Descripción	Si no existe la palabra clave indicada en la palabra de inicio o en la palabra de fin, se debe mostrar un error y todo el proceso de ejecución se debe detener.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 70: Requisito no funcional RNF-017.

Código	RNF-018		
Nombre	Modelo estadístico: exclusividad.		
Descripción	Se debe poder elegir únicamente un modelo estadístico de entre los dos disponibles: frecuencia por nivel o frecuencia por nodo.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 71: Requisito no funcional RNF-018.

Código	RNF-019		
Nombre	Formato de exportación.		
Descripción	Ambas opciones, PDF o Gephi deben poder ser elegidas como formato de exportación del trie resultante. No existe exclusividad entre ellas.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 72: Requisito no funcional RNF-019.

Código	RNF-020		
Nombre	Previsualización de color: propiedades gráficas y métricas.		
Descripción	Al lado de los botones de color para la coloración debe existir un campo no editable que muestra el color seleccionado. Por defecto, este campo será de color negro.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 73: Requisito no funcional RNF-020.

Código	RNF-021		
Nombre	Métricas: formato frecuencias estadísticas.		
Descripción	Si se selecciona la métrica de frecuencias estadísticas bajo un umbral, el campo de texto del umbral debe aceptar únicamente dígitos numéricos formando números comprendidos entre 0 y 1, ambos incluidos.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 74: Requisito no funcional RNF-021.

Código	RNF-022		
Nombre	Métricas: formato frecuencias de inserción.		
Descripción	Si se selecciona la métrica de frecuencias de inserción bajo un umbral, el campo de texto del umbral debe aceptar únicamente dígitos numéricos formando números enteros y positivos.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 75: Requisito no funcional RNF-022.

Código	RNF-023		
Nombre	Representación gráfica: grosor de enlaces.		
Descripción	En la representación gráfica, el grosor de los enlaces está directamente relacionada con el peso que tiene el enlace. A mayor peso, mayor grosor. El peso de los enlaces se establece como la etiqueta del enlace.		
Prioridad	Alta	Verificabilidad	Alta

Tabla 76: Requisito no funcional RNF-023.

Código	RNF-024		
Nombre	Concurrencia entre proceso backend e interfaz gráfica.		
Descripción	La interfaz se ejecuta en el hilo principal, mientras que el proceso backend se ejecuta en un nuevo hilo de ejecución. Ambos deben ser concurrentes para que la interfaz se actualice correctamente (escritura en el área Log).		
Prioridad	Alta	Verificabilidad	Alta

Tabla 77: Requisito no funcional RNF-024.

3.6. Tecnologías utilizadas

En esta sección se describen todas las tecnologías utilizadas y el porqué de su uso en este proyecto.

3.6.1. Java

Java es el nombre del lenguaje de programación desarrollado por James Gosling y su equipo de *Sun Microsystems* en 1991. En 1996 se produjo el primer lanzamiento de los que se denominaría JDK 1.0 o *Java Development Kit* 1.0.

La versión más reciente de Java es JDK 8. El lenguaje lleva siendo desarrollado diecinueve años y su rendimiento y aplicaciones han crecido tanto que prácticamente se utiliza en cualquier parte [67]. Actualmente Java es la base para prácticamente todas las aplicaciones web y está siendo utilizado como base fundamental del sistema operativo Android (uno de los sistemas operativos móviles más distribuidos hoy en día).

En 2009 Oracle adquirió *Sun Microsystems*, por lo que Java ahora es propiedad de *Oracle*.

3.6.1.1. Funcionamiento

La sintaxis de Java proviene de una mezcla de C y C++ y se trata de un lenguaje compilado, lo que significa que puede ejecutarse en cualquier Máquina Virtual de Java (*JVM* por sus siglas en Inglés), independientemente de la arquitectura de la máquina física en la que se ejecuta la máquina virtual [68]. Esto se consigue gracias a que el formato de salida del compilador de Java no es código ejecutable, sino *Bytecode*. El *Bytecode* es un conjunto de instrucciones altamente optimizado que está diseñado para ejecutarse en un sistema que ejecuta Java, el cual es llamado Máquina Virtual de Java. Esto quiere decir que la Máquina Virtual de Java es un intérprete de *Bytecode*.

Java es un lenguaje de programación orientado a objetos. Este tipo de paradigma de programación se basa en las interacciones entre objetos. Con el uso de objetos se especifican el tipo de los elementos que usan y su manejo se simplifica enormemente. Este paradigma ofrece las siguientes tres características: encapsulación, herencia y polimorfismo. La encapsulación permite unir el código y los datos que manipula y mantenerlos seguros e inalterables desde otros puntos del código. La encapsulación es el fundamente de lo que se denomina objeto. La herencia es la característica de los objetos que permite adquirir a un objeto las propiedades de otro. El polimorfismo ayuda a reducir las implementaciones necesarias para manejar diferentes tipos de datos. Un ejemplo de polimorfismo es la implementación de una pila de datos. Esta pila puede contener diferentes tipos de objetos, pero su manipulación es la misma para todos ellos.

3.6.1.2. Aplicaciones

Java ha potenciado enormemente el desarrollo de aplicaciones web. Java está presente en dos tipos de aplicaciones web: *Applets* y *Servlets*.

Las *Applets* son pequeñas aplicaciones web implementadas en Java y diseñadas para ejecutarse en un navegador web. Comúnmente se utilizan para mostrar información proporcionada por un servidor, para el manejo de información introducida por un usuario o para cargar simples funcionalidades independientes de un servidor, como por ejemplo una calculadora.

Los *Servlets* son aplicaciones Java presentes en un servidor. Estas aplicaciones se encargan de proporcionar respuestas a las peticiones que le llegan al servidor. Los *Servlets* se utilizan como componentes adicionales a las aplicaciones alojadas en servidores web. El uso más común es el de generar páginas web de forma dinámica en base a los parámetros de petición que llegan al servidor que contiene los *Servlets*.

Por otra parte, Java se utiliza para programar todo tipo de aplicaciones autónomas. Este tipo de aplicaciones pueden ser autocontenidas (gracias a las librerías de Java) o conectarse con otras aplicaciones autónomas. Además, para estas aplicaciones autónomas Java posee una librería especial denominada *Java Swing* que permite desarrollar interfaces gráficas.

Tal y como se ha mencionado anteriormente, Java es uno de los lenguajes base de programación de Android. Concretamente, Java se encarga de toda la interfaz gráfica del sistema operativo. Las aplicaciones móviles de Android se escriben utilizando Java, enfocado al desarrollo móvil, donde existen librerías específicas para el desarrollo hacia dispositivos móviles. Se debe mencionar que en Android, las aplicaciones Java no corren directamente sobre una Máquina Virtual de Java, sino que el *Bytecode* resultante es compilado en un ejecutable Dalvik y posteriormente ejecutado en una Máquina Virtual Dalvik (se trata de una Máquina Virtual especializada, diseñada específicamente para Android).

3.6.1.3. Ventajas

- Portabilidad: gracias a que Java es ejecutable independiente de la plataforma, es posible escribir software en una plataforma y luego distribuirlo masivamente sin importar de las especificaciones de la plataforma destino.
- Código abierto: Java es software libre por lo que existe una gran comunidad de desarrolladores que comparten sus conocimientos en libros e Internet. La literatura de Java está ampliamente extendida y es enseñada en muchos niveles de la educación pública y privada. Al ser software libre los usuarios tienen la libertad de distribuir, copiar, modificar, mejorar y ejecutar el software. Además, se trata de un software gratuito. Java no precisa de una licencia de desarrollo concreta por lo que el desarrollo de software autónomo es gratuito. Se debe tener en cuenta que para formar parte de la comunidad de desarrollo de aplicaciones para Android se debe tener una licencia denominada *Android Developer*, que cuesta unos veinticinco dólares americanos.
- Java Swing: Java Swing es la librería de Java encargada de generar elementos gráficos en una aplicación. Con ella se diseñan interfaces gráficas que simplifican el uso de una aplicación por parte del usuario.

3.6.2. UML

UML, por sus siglas en Inglés *Unified Modeling Language*, es un lenguaje de modelado de sistemas software creado a principios de 1997.

Este lenguaje es una mezcla de diversas metodologías de modelado de sistemas y surgió como una mezcla de ellos para crear un lenguaje de modelado de sistemas universal. Se basa en las metodologías OMT (*Objetct-Modeling Technique*) de James Rumbaugh, Booch de Grady Booch y OOSE (*Objetct-Oriented Software Engineering*) de Ivar Jackobson. Los tres autores trabajaban para la empresa *Rational Software Corporation*, que en 1996 les propuso crear un lenguaje de modelado universal debido a la gran cantidad de lenguajes de modelado que estaba surgiendo.

Ese mismo año, los Tres Amigos, como les llamaban por sus continuas discusiones sobre el tema, presentaron su propuesta al *Object Management Group* como metodología de estandarización de modelado de sistemas. En 1997 finalmente se aceptó su propuesta y desde entonces el modelado de lenguajes ha sido predominado por UML.

En 2005, UML fue adoptado como estándar por la Organización Internacional de Normalización o ISO por sus siglas en Inglés.

Este lenguaje ha ido evolucionando desde los noventa hasta hoy, y en 2015 fue propuesta la versión 2.5, que hoy en día es la versión más actual [69].

3.6.2.1. Aplicaciones

Con el uso de este lenguaje se pueden modelar los componentes de un sistema (y su interacción entre ellos), el funcionamiento general del sistema y cómo interaccionan los componentes e interfaces con las que el sistema cuenta.

Todo esto se realiza mediante diferentes tipos de diagramas, cada uno con un propósito específico. En este proyecto se han utilizado:

- Diagrama de componentes: para representar la arquitectura de Modelo Vista Controlador que sigue el sistema.
- Diagrama de casos de uso: para representar y generar las interacciones que los usuarios pueden hacer y los respectivos requisitos que el sistema debe seguir para satisfacer los objetivos del proyecto.

3.6.2.2. Ventajas

- Se trata del lenguaje de modelado de sistemas más popular de hoy en día y sus metodologías y métodos de descripción de sistemas están ampliamente documentados, tanto vía online, como en libros de texto.
- Gracias a la gran cantidad de diagramas con las que cuenta el lenguaje, se puede describir por completo el sistema implementado.
- Se ha utilizado previamente durante los estudios universitarios, por lo que su puesta en práctica es ya conocida.

3.7. Software utilizado

En esta sección del documento se describe todo el software utilizado para la realización del proyecto.

3.7.1. Eclipse

Eclipse es un programa informático que permite la programación de proyectos software en múltiples lenguajes de programación y múltiples plataformas.

Eclipse empezó como un proyecto de IBM Canadá para reemplazar a otro proyecto que ya tenían. En 2001 se formó un consorcio entre los principales líderes de la industria (IBM, Red Hat, etc.) para crear la junta de eclipse.org [70]. En 2004 se creó la Fundación Eclipse, que actualmente lidera el desarrollo de Eclipse. Esta fundación se estableció sin ánimo de lucro y desde entonces es uno de los principales entornos de desarrollo de programas Java.

Eclipse fue desarrollado mayoritariamente en Java, aunque otros lenguajes de programación como ANSI C y C++, entre otros fueron utilizados.

3.7.1.1. Funcionalidades

Eclipse se caracteriza por poseer un gran número de *plugins* o funcionalidades adicionales que le permiten agregar nuevas funcionalidades a las ya existentes. Esto le permite operar con diversos lenguajes de programación, incluir herramientas de monitorización de rendimiento, agregar opciones y menús contextuales para facilitar la labor del desarrollador, etc.

Eclipse posee varias plataformas de desarrollo en función del objetivo a desarrollar:

- Plataforma de Cliente Enriquecido o RCP por sus siglas en Inglés: es la plataforma para desarrollo de aplicaciones genéricas. El componente de esta plataforma utilizado para el desarrollo del código del proyecto es Eclipse Workbench.
- Plataforma Servidor: es la plataforma que soporta *Tomcat* y *GlassFish*, entre otros, y se utiliza para el desarrollo de entornos y aplicaciones de servidor.
- Plataforma de Herramientas Web o WTP por sus siglas en Inglés: esta plataforma se encarga de gestionar todos los proyectos de la Fundación Eclipse. Dentro de los diferentes proyectos se encuentran las Herramientas de Modelado, entre las cuales se encuentra UML.

Uno de los *plugins* más utilizados en Eclipse es *Android Development Tools* (o ADT por sus siglas en Inglés), desarrollado por Google y con el objetivo de extender las capacidades de Eclipse para permitir a los desarrolladores crear y manejar proyectos Android.

3.7.1.2. Ventajas

- Software libre y gratuito.
- Gran comunidad de desarrolladores que han creado un amplio abanico de documentación y resolución de preguntas por Internet en el que el programador se puede apoyar en caso de necesitarlo.
- Amplia red de plugins desarrollados exclusivamente para Eclipse y que facilitan la labor al desarrollador. En este proyecto se ha utilizado el diseñador gráfico de modelo "*drag & drop*" de interfaces de usuario *Swing Designer*.
- Previo conocimiento y uso de este entorno de desarrollo a lo largo de la carrera universitaria y el marco laboral en el que he estado presente.

3.7.2. Gephi

Gephi es un paquete software de código abierto destinado al análisis y visualización de redes. El software fue desarrollado por alumnos de la Universidad Tecnológica de Compiègne, Francia, en el año 2008.

El proyecto Gephi ha participado varias veces en el *Google Summer of Code*, en los años 2009, 2010, 2011, 2012 y 2013. Se trata de un programa anual que premio a los estudiantes mayores de 19 años que completan un proyecto de software libre en ese mismo verano.

Gracias a su arquitectura y funcionalidades, Gephi ha sido utilizado en numerosos estudios académicos y periodísticos, como por ejemplo [71] y [72], y gracias a su naturaleza puede ser utilizado para cualquier estudio que se centre en la teoría de grafos.

La última versión de Gephi se publicó en 2013 y desde entonces no se han liberado nuevas versiones.

3.7.2.1. Funcionalidades

Este software permite revelar intuitivamente patrones y líneas de tendencia en datos gracias a que utiliza un motor gráfico 3D para representar gráficamente grafos grandes y en tiempo real.

La arquitectura de Gephi permite explorar, analizar, colocar, filtrar, clusterizar, manipular y exportar todo tipo de redes [73]. Trabaja con una extensión del formato XML donde se representan las coordenadas y los atributos de todos los nodos y en enlaces del grafo.

Con Gephi se pueden crear, importar y exportar grafos. En cuanto al análisis de los datos, Gephi posee múltiples métricas sobre análisis de grafos, como por ejemplo Centralidad, Grado, etc. Por otra parte, los elementos del grafo, nodos y enlaces, se pueden recolocar, cambiar de color, cambiar tamaño, asignarles etiquetas y pesos. Esto permite al analista visualizar fácilmente nodos de interés o patrones en los nodos y las relaciones entre ellos.

Gephi posee diferentes algoritmos de análisis de datos permitiendo posicionar, redimensionar y colorear nodos y enlaces en función de ciertos parámetros especificados por un analista.

Gephi cuenta además con un laboratorio de análisis que permite mostrar y filtrar los nodos y enlaces por cierto criterio que el analista especifica. De esta forma se pueden con detalle las características de los elementos “interesantes”.

Al ser un proyecto de código abierto, Gephi tiene una gran comunidad de desarrolladores y usuario que contribuyen a expandir el software existente, mediante el desarrollo de *plugins* y la resolución de dudas y problemas.

Por otra parte, además de la aplicación de escritorio, Gephi posee API de Java, con la que todas las funcionalidades de la aplicación de escritorio se pueden implementar en un programa Java independiente.

3.7.2.2. Ventajas

- Extenso conjunto de algoritmos y métricas para el análisis de patrones.
- La existencia de una API de Java para implementar las funcionalidades del software de escritorio, además de poder crear y exportar el trie generado gráficamente.
- Amplia documentación y diversos foros de usuarios y desarrolladores en cuanto a la utilización del API de Gephi.
- Familiarización previa del software, ya que Gephi se ha utilizado en proyectos anteriores.

3.7.3. StarUML 2

StarUML surgió como herramienta de modelado UML en 2014 por la compañía *MKLab*. En sus primeras versiones (StarUML 1) era una herramienta de software libre, hasta que en su segunda versión (StarUML) que fue reescrita por completo, se adoptó una licencia propietaria debido a la escasez de fondos que la empresa tenía para mantener y desarrollar el software.

Según la propia página oficial de la compañía, StarUML es una de las herramientas más populares de modelado UML, con más de cuatro millones de descargas en más de ciento cincuenta países.

Actualmente su última versión es 2.5.0 y tiene un período indefinido de evaluación gratuita [74].

3.7.3.1. Funcionalidades

StarUML 2 ofrece once diferentes tipos de diagramas de modelado UML, entre los que se encuentran los necesarios para la definición de este proyecto (diagrama de clases, diagrama de componentes y diagrama de casos de uso).

Por otra parte, posee extensiones para diferentes lenguajes de programación, de forma que se pueda ser integrado en diferentes plataformas.

3.7.3.2. Ventajas

- Se trata de un software ampliamente extendido, que cuenta con extensa documentación y gran número de foros de debate online.
- Software que cuenta con una licencia gratuita de evaluación, con la que se pueden desarrollar todos los diagramas necesarios.
- Uso previo del software, por lo que su utilización no requiere de aprendizaje extra.

Capítulo 4: Experimentación

En este capítulo, en primer lugar, se presenta toda la experimentación realizada con el sistema implementado. Esta experimentación se divide en tres fases: Fase de pruebas, Fase de resultados y Fase de evaluación.

En la primera fase se detalla el dominio de los datos utilizados y las configuraciones de los experimentos realizados. En la segunda fase se presentan los resultados obtenidos tras realizar la batería de pruebas de la primera fase. Por último, en la tercera fase se interpretan los resultados obtenidos de la experimentación.

4.1. Fase de Pruebas

En primer lugar se describe el dominio de los datos utilizados para la experimentación. El dominio de datos utilizado está definido por la interacción de diversos usuarios con una terminal de comandos del sistema operativo UNIX. Con el estudio y análisis de los comandos que un usuario introduce en la consola UNIX se puede modelar a un usuario con el objetivo de:

- Determinar su nivel de conocimiento del sistema operativo, analizando los directorios y ficheros a los que accede.
- Determinar su nivel de experto como usuario de la consola de UNIX, analizando la longitud y complejidad de los comandos que realiza para moverse dentro del sistema operativo.
- Determinar si el usuario expresa intenciones maliciosas, analizando los directorios y fichero a los que accede y las operaciones que realiza sobre éstos.

En cuanto al conjunto de datos utilizados en la experimentación, se compone de los comandos introducidos por 50 usuarios durante un período de tiempo. Estos datos han sido utilizados previamente por [75] con el objetivo de detectar intrusos en un sistema. En ese estudio se intercalan los comandos introducidos por usuarios con aquellos que suponen una intrusión en el sistema. En esta experimentación se ha cogido los datos originales de los 50 usuarios. Por cada usuario existe un fichero de 15.000 comandos. Los datos están disponibles en la página web de Matt Schonlau [76].

De los 15.000 comandos por usuario originales, se han escogido 5000 comandos y de los 50 usuarios originales se han escogido 5. El objetivo de esta experimentación es demostrar la funcionalidad del software, por lo que el tamaño o el número de usuarios escogidos no es determinante para realizar la evaluación del sistema implementado.

Las pruebas sobre los datos escogidos se van a realizar en 3 configuraciones de profundidad: 3, 5 y 7. Se han escogido estas profundidades debido a que profundidades mayores, el tiempo de creación de los tries es muy elevado. Para cada profundidad se van a obtener los valores de soporte que miden la relevancia del cada evento, o en este caso, la relevancia de cada comando.

A continuación se muestra una ejecución de prueba para mostrar tanto la configuración de la interfaz gráfica de la herramienta, como del trie que genera. En esta ejecución se han utilizado los datos de ejemplo utilizados en la sección *2.4.1.2*, para la explicación de las fases de construcción de un trie.

La configuración de la herramienta es:

- Método de segmentación: Segmentación por profundidad, de profundidad 3.
- Modelo estadístico: Valores de soporte.

- Formato de exportación: PDF.
- Propiedades gráficas: Etiquetas de enlaces.
- Métricas:
    - Frecuencias estadísticas: umbral de 0.3 en color verde.
    - Frecuencias de inserción: umbral de 3 en color azul.

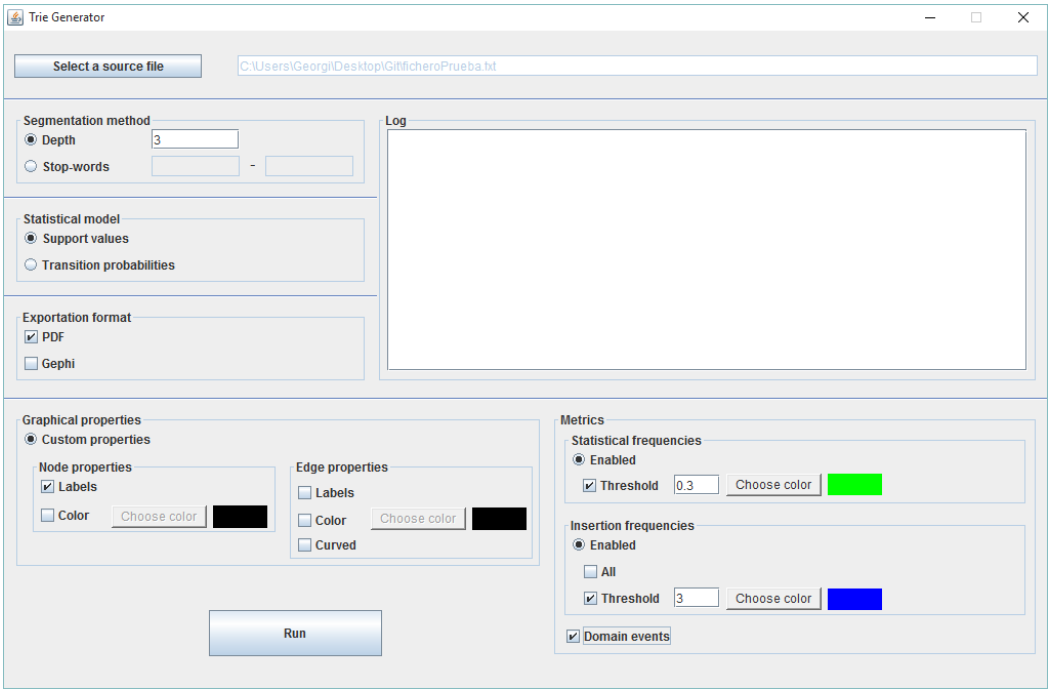

Ilustración 16: Ejecución de ejemplo.

El trie resultante, tras abrirlo y editarlo con la aplicación de escritorio de Gephi, es el siguiente:

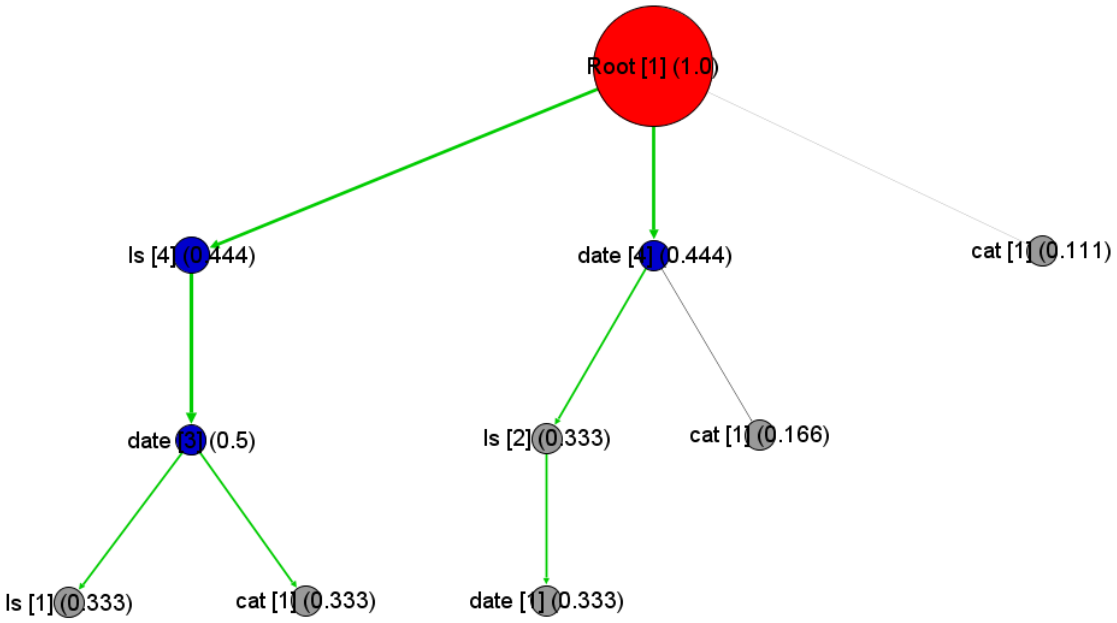

Ilustración 17: Representación gráfica del trie resultante tras la ejecución de ejemplo.

A continuación se presenta la configuración de las pruebas realizadas:

Configuraciones de pruebas				
Usuario	Profundidad	Tiempo (min)	Número de nodos	Número de enlaces
1	3	0:41	1635	1634
1	5	8:50	4417	4416
1	7	17:23	8140	8139
2	3	0:31	1036	1035
2	5	6:37	2851	2850
2	7	17:10	5356	5355
3	3	0:48	1045	1044
3	5	15:35	2986	2985
3	7	17:12	5607	5606
4	3	1:75	1472	1471
4	5	17:15	4081	4080
4	7	19:11	7639	7638
5	3	0:33	1057	1056
5	5	7:43	2894	2893
5	7	21:26	5440	5439

Tabla 78: Tabla de configuración de experimentos.

Como puede observarse, a medida que se aumenta la profundidad, el tiempo de creación del trie y sus componentes aumentan.

4.2. Fase de Resultados

En esta fase se muestran y evalúan los resultados obtenidos de la Fase de Pruebas.

Para evaluar los resultados, se han escogido los 20 comandos que más valor de soporte tienen de cada profundidad y de cada usuario. Así, se puede considerar que ese conjunto de comandos representa a cada usuario.

Con la finalidad de mostrar estos comandos de la forma más gráfica posible, los resultados de cada usuario se muestran mediante una tabla y una gráfica de barras asociada a dicha tabla. Se han omitido las representaciones gráficas de los tries obtenidos, ya que éstos poseen miles de nodos y enlaces, que en una hoja de tamaño A4 forman un cúmulo indistinguible de nodos y enlaces. Sin embargo, la herramienta Gephi (en su versión de aplicación de escritorio) utilizada en la herramienta desarrollada, permite seleccionar determinados enlaces y nodos, pudiendo así explorar de forma precisa cada uno de los tries obtenidos.

En esta fase se van a presentar los resultados obtenidos para el primero de los cinco usuarios. El resto de los resultados del resto de usuarios se presenta en el *Anexo I*.

Profundidad 3	
Valor de soporte	Nodo
0,278	Root-netscape
0,252	Root-netscape-netscape
0,23	Root-netscape-netscape-netscape
0,078	Root-sh
0,043	Root-csh
0,042	Root-egrep
0,04	Root-sendmail
0,033	Root-egrep-egrep
0,03	Root-cat
0,029	Root-launchef
0,025	Root-egrep-egrep-egrep
0,024	Root-expr
0,023	Root-generic
0,023	Root-ls
0,016	Root-MediaMai
0,016	Root-sendmail-sendmail
0,015	Root-launchef-sh
0,015	Root-rm
0,015	Root-sh-MediaMai
0,015	Root-java

Tabla 79: Tabla de valores de soporte para Usuario 1 con Profundidad 3.

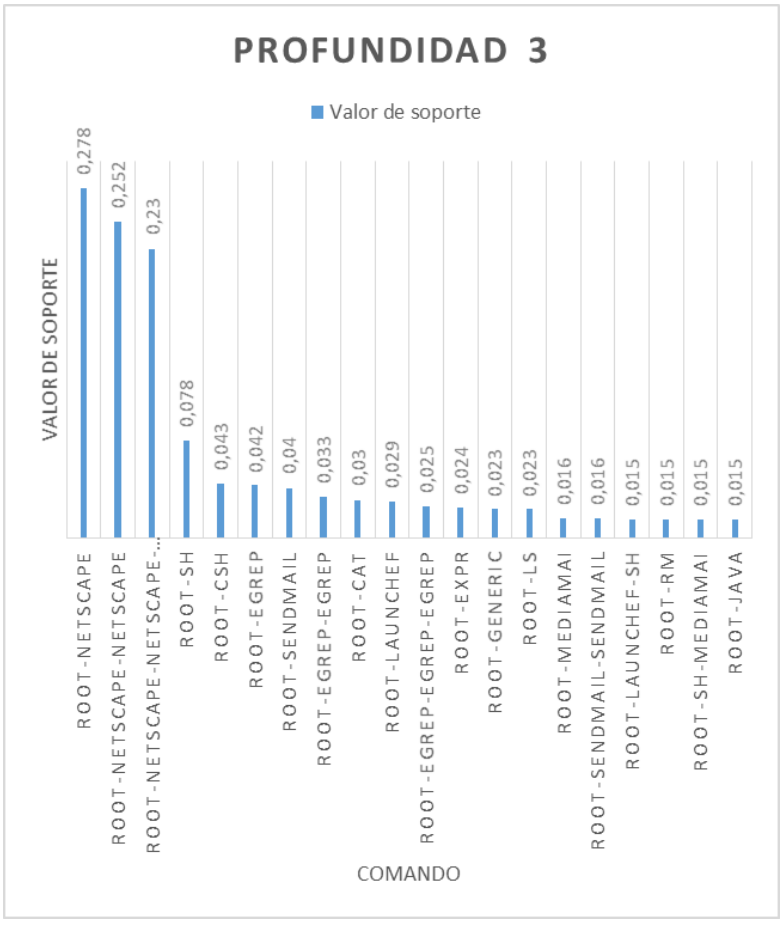

Ilustración 18: Gráfica de valores de soporte para Usuario 1 con Profundidad 3.

Profundidad 5	
Valor de soporte	Nodo
0,278	Root-netscape
0,252	Root-netscape-netscape
0,23	Root-netscape-netscape-netscape
0,212	Root-netscape-netscape-netscape-netscape
0,195	Root-netscape-netscape-netscape-netscape-netscape
0,078	Root-sh
0,043	Root-csh
0,042	Root-egrep
0,04	Root-sendmail
0,033	Root-egrep-egrep
0,03	Root-cat
0,029	Root-launchef
0,025	Root-egrep-egrep-egrep
0,024	Root-expr
0,023	Root-generic
0,023	Root-ls
0,016	Root-MediaMai
0,016	Root-sendmail-sendmail
0,016	Root-egrep-egrep-egrep-egrep
0,015	Root-launchef-sh

Tabla 80: Tabla de valores de soporte para Usuario 1 con Profundidad 5.

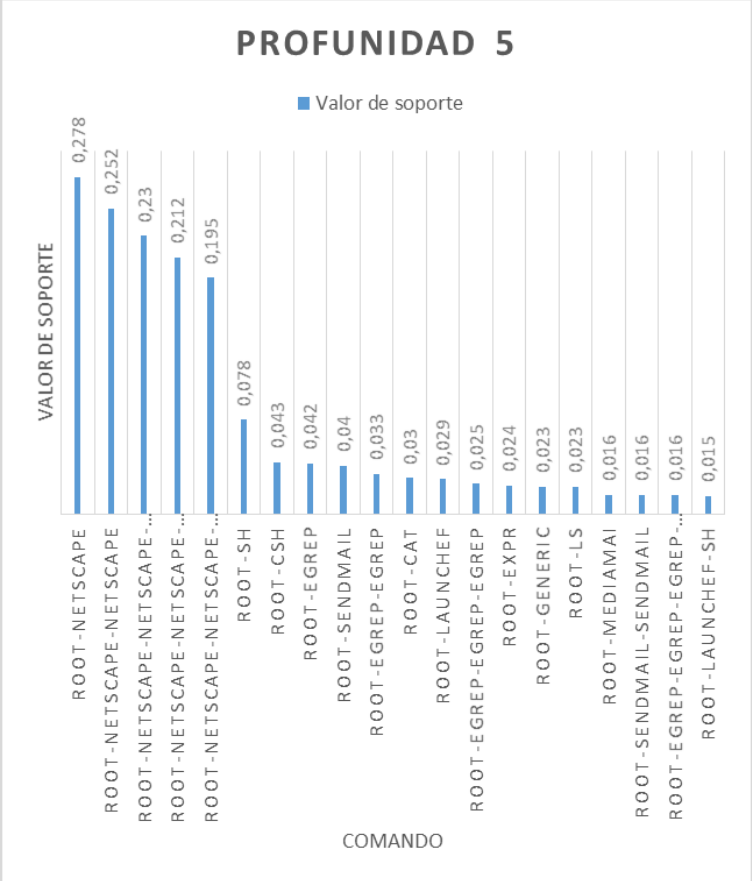

Ilustración 19: Gráfica de valores de soporte para Usuario 1 con Profundidad 5.

Profundidad 7	
Valor de soporte	Nodo
0,278	Root-netscape
0,252	Root-netscape-netscape
0,23	Root-netscape-netscape-netscape
0,212	Root-netscape-netscape-netscape-netscape
0,196	Root-netscape-netscape-netscape-netscape-netscape
0,18	Root-netscape-netscape-netscape-netscape-netscape-netscape
0,165	Root-netscape-netscape-netscape-netscape-netscape-netscape-netscape
0,078	Root-sh
0,043	Root-csh
0,042	Root-egrep
0,04	Root-sendmail
0,033	Root-egrep-egrep
0,03	Root-cat
0,029	Root-launchef
0,024	Root-expr
0,024	Root-egrep-egrep-egrep
0,023	Root-generic
0,023	Root-ls
0,016	Root-MediaMai
0,016	Root-sendmail-sendmail

Tabla 81: Tabla de valores de soporte para Usuario 1 con Profundidad 7.

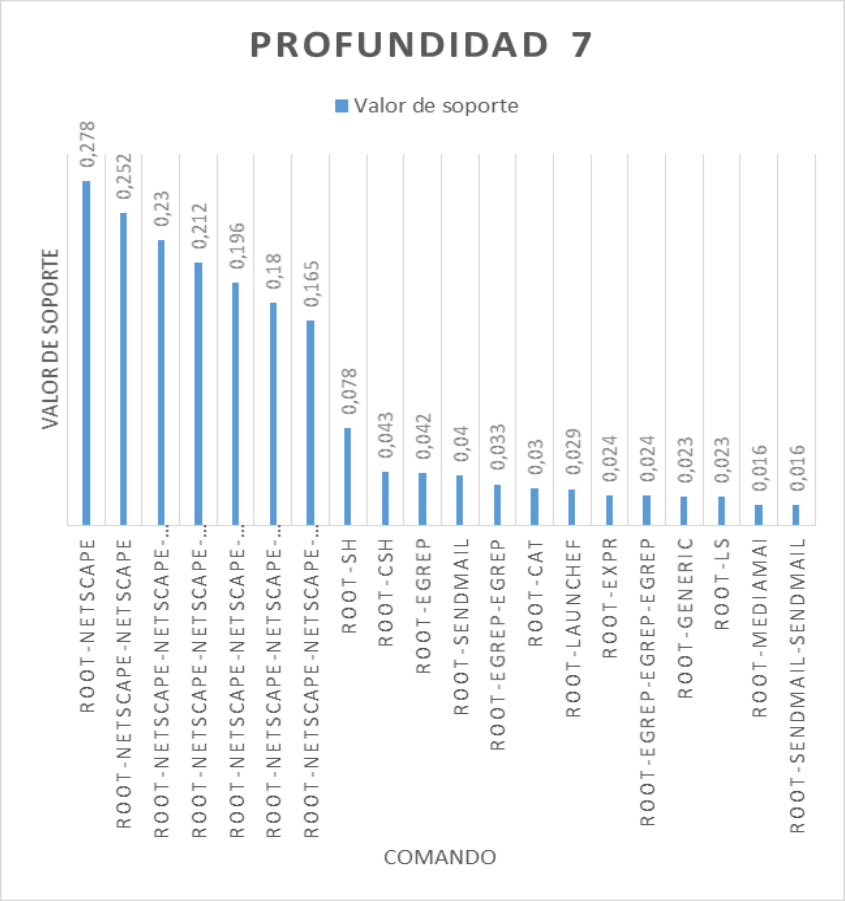

Ilustración 20: Gráfica de valores de soporte para Usuario 1 con Profundidad 7.

4.3. Fase de Evaluación

Tras la realización de la batería de pruebas, los resultados obtenidos pueden ser interpretados de la siguiente manera: dado que el valor de soporte mide la relevancia de una secuencia de eventos en un nivel de profundidad, se pueden extraer los comandos más relevantes de cada nivel de profundidad. A continuación se presentan los comandos más relevantes de los tres primeros niveles de cada usuario:

Usuario	Nivel	Comando más relevante	Valor de soporte
1	1	netscape	0,278
	2	netscape-netscape	0,252
	3	netscape-netscape-netscape	0,230
2	1	gcc	0,138
	2	uname-nawk	0,069
	3	uname-nawk-cpp	0,053
3	1	egrep	0,174
	2	egrep-egrep	0,130
	3	egrep-egrep-egrep	0,087
4	1	sh	0,204
	2	more-sh	0,099
	3	sh-more-sh	0,089
5	1	generic	0,112
	2	date-generic	0,032
	3	ls-sed-FIFO	0,022

Tabla 82: Tabla con comandos más relevantes de cada usuario.

Se puede observar que con el cambio de profundidad, los comandos más relevantes no siempre son los comandos sucesivos del nivel anterior. Esto se produce en los usuarios 2, 4 y 5, donde en el nivel 2 de profundidad, el comando más relevante posee un comando anterior diferente del obtenido en la profundidad 1. Por ejemplo, del usuario 2, el comando más relevante de profundidad 2 está formado por la secuencia *uname-nawk*, mientras que su comando más relevante de profundidad 1 es *gcc*. Por este motivo, varios valores de profundidad diferentes proporcionan información diversa sobre los usuarios estudiados. Con esto se puede interpretar que no es necesario seleccionar siempre una única profundidad de segmentación para un determinado caso.

Los usuarios 1 y 3 muestran que sí se mantiene la secuencia de comandos con el comando más relevante en cada profundidad. Por ejemplo, el usuario 1 muestra que su comando más relevante de profundidad 1 es *netscape*, y los comandos más relevantes de las profundidades 2 y 3 se forman en secuencia con éste.

Por otra parte, los resultados revelan que los cinco usuarios estudiados no comparten similitudes respecto al uso de la terminal de comandos de UNIX, ya que sus comandos más relevantes no se parecen. Esto demuestra que los usuarios han utilizado la consola con diferentes finalidades.

Por último, se observa que los valores de soporte de los comandos más relevantes son muy bajos. Esto se debe a que el número de comandos de esa profundidad es muy grande respecto a la frecuencia de inserción de cada comando. Se debe recordar que el valor de soporte se calcula como:

$$soporte(x_i) = \frac{\text{frecuenciaInserción}(x_i)}{\sum_1^n \text{frecuenciaInserción}(x_{n,i})}$$

Ecuación 4: Fórmula para calcular el valor de soporte.

, donde *i* representa el nivel de profundidad y *n* cada nodo de esa profundidad. Si el denominador es demasiado grande respecto al numerador, el valor de soporte tiende a 0. Esto se traduce en que existe una gran variedad de comandos de la misma longitud que los usuarios introducen a la hora de utilizar las terminales de UNIX.

Capítulo 5: Desarrollo del Proyecto

5.1. Planificación

La planificación del desarrollo del proyecto se ha separado en las siguientes fases:

- Fase de planificación: fase inicial de una tarea en la que una se estudian y analizan los aspectos de la tarea de forma que pueda ser implementada en el sistema.
- Fase de programación: fase intermedia de una tarea en la que se implementa y prueba la tarea en el sistema.
- Fase de documentación: fase final de una tarea en la que se documenta todo lo relevante de la tarea, tanto en el código fuente, como en la memoria del proyecto.

La siguiente ilustración muestra la planificación del proyecto:

Fecha	Semana	Tareas
02/02/2015 - 08/02/2015	6	Realizar cronograma del proyecto.
09/02/2015 - 15/02/2015	7	Leer y entender la documentación respecto a los objetivos del proyecto.
16/02/2015 - 22/02/2015	8	Leer y entender la documentación respecto a los objetivos del proyecto.
23/02/2015 - 01/03/2015	9	Fase de planificación: segmentación.
02/03/2015 - 08/03/2015	10	Fase de programación: segmentación.
09/03/2015 - 15/03/2015	11	Fase de programación: segmentación.
16/03/2015 - 22/03/2015	12	Fase de planificación: almacenamiento del trie.
23/03/2015 - 29/03/2015	13	Fase de programación: almacenamiento del trie.
30/03/2015 - 05/04/2015	14	Fase de planificación: modelo estadístico.
06/04/2015 - 12/04/2015	15	Fase de programación: modelo estadístico.
13/04/2015 - 19/04/2015	16	Fase de programación: modelo estadístico.
20/04/2015 - 26/04/2015	17	Fase de planificación: integración API Gephi.
27/04/2015 - 03/05/2015	18	Fase de programación: representación gráfica del trie.
04/05/2015 - 10/05/2015	19	Fase de planificación: organización de clases.
11/05/2015 - 17/05/2015	20	Fase de programación: flujo de ejecución mejorado.
18/05/2015 - 24/05/2015	21	Fase de programación: ejecución parametrizada por teclado.
25/05/2015 - 31/05/2015	22	Fase de documentación: comentar código implementado.
01/06/2015 - 07/06/2015	23	Fase de planificación: diseño de interfaz gráfica y funcionalidades base.
08/06/2015 - 14/06/2015	24	Fase de programación: primer prototipo de interfaz con funcionalidades base.
15/06/2015 - 21/06/2015	25	Fase de programación: ampliación de funcionalidades de la interfaz.
22/06/2015 - 28/06/2015	26	Fase de programación: ampliación de funcionalidades de la interfaz.
29/06/2015 - 05/07/2015	27	Fase de programación: parametrización de pintado e inserción de métricas.
06/07/2015 - 12/07/2015	28	Fase de programación: parametrización de pintado e inserción de métricas.
13/07/2015 - 19/07/2015	29	Fase de documentación: comentar código implementado. Fase de programación: Prototipo final del software.
20/07/2015 - 26/07/2015	30	Fase de documentación: comienzo de memoria del proyecto.
27/07/2015 - 02/08/2015	31	Fase de documentación: ampliación de la memoria del proyecto.
03/08/2015 - 09/08/2015	32	Fase de documentación: ampliación de la memoria del proyecto.
10/08/2015 - 16/08/2015	33	Fase de documentación: ampliación de la memoria del proyecto.
17/08/2015 - 23/08/2015	34	Fase de documentación: ampliación de la memoria del proyecto.
24/08/2015 - 30/08/2015	35	Fase de programación: prototipo final.
31/08/2015 - 06/09/2015	36	Fase de documentación: ampliación de la memoria del proyecto.
07/09/2015 - 13/09/2015	37	Fase de documentación: ampliación de la memoria del proyecto.
14/09/2015 - 20/09/2015	38	Fase de documentación: ampliación de la memoria del proyecto.
21/09/2015 - 27/09/2015	39	Fase de documentación: memoria final.

Tabla 83: Tabla con la planificación del proyecto.

En la siguiente ilustración se representa la planificación del proyecto mediante un diagrama de Gantt

Nombre	Fecha de inicio	Fecha de fin
Realizar cronograma del proyecto	2/02/15	8/02/15
Leer y entender la documentación respecto a los objetivos del proyecto	9/02/15	22/02/15
Fase de planificación: segmentación	23/02/15	1/03/15
Fase de programación: segmentación	2/03/15	15/03/15
Fase de planificación: almacenamiento del trie	16/03/15	22/03/15
Fase de programación: almacenamiento del trie	23/03/15	29/03/15
Fase de planificación: modelo estadístico	30/03/15	5/04/15
Fase de programación: modelo estadístico	6/04/15	19/04/15
Fase de planificación: integración API Gephi	20/04/15	26/04/15
Fase de programación: representación gráfica del trie	27/04/15	3/05/15
Fase de planificación: organización de clases	4/05/15	10/05/15
Fase de programación: flujo de ejecución mejorado	11/05/15	17/05/15
Fase de programación: ejecución parametrizada por teclado	18/05/15	24/05/15
Fase de documentación: comentar código implementado	25/05/15	31/05/15
Fase de planificación: diseño de interfaz gráfica y funcionalidades base	1/06/15	7/06/15
Fase de programación: primer prototipo de interfaz con funcionalidades base	8/06/15	14/06/15
Fase de programación: ampliación de funcionalidades de la interfaz	15/06/15	28/06/15
Fase de programación: parametrización de pintado e inserción de métricas	29/06/15	12/07/15
Fase de documentación: comentar código implementado	13/07/15	19/07/15
Fase de programación: prototipo final del software	13/07/15	30/08/15
Fase de documentación: comienzo de memoria del proyecto	20/07/15	26/07/15
Fase de documentación: ampliación de la memoria del proyecto	27/07/15	20/09/15
Fase de documentación: acabado memoria final	21/09/15	27/09/15

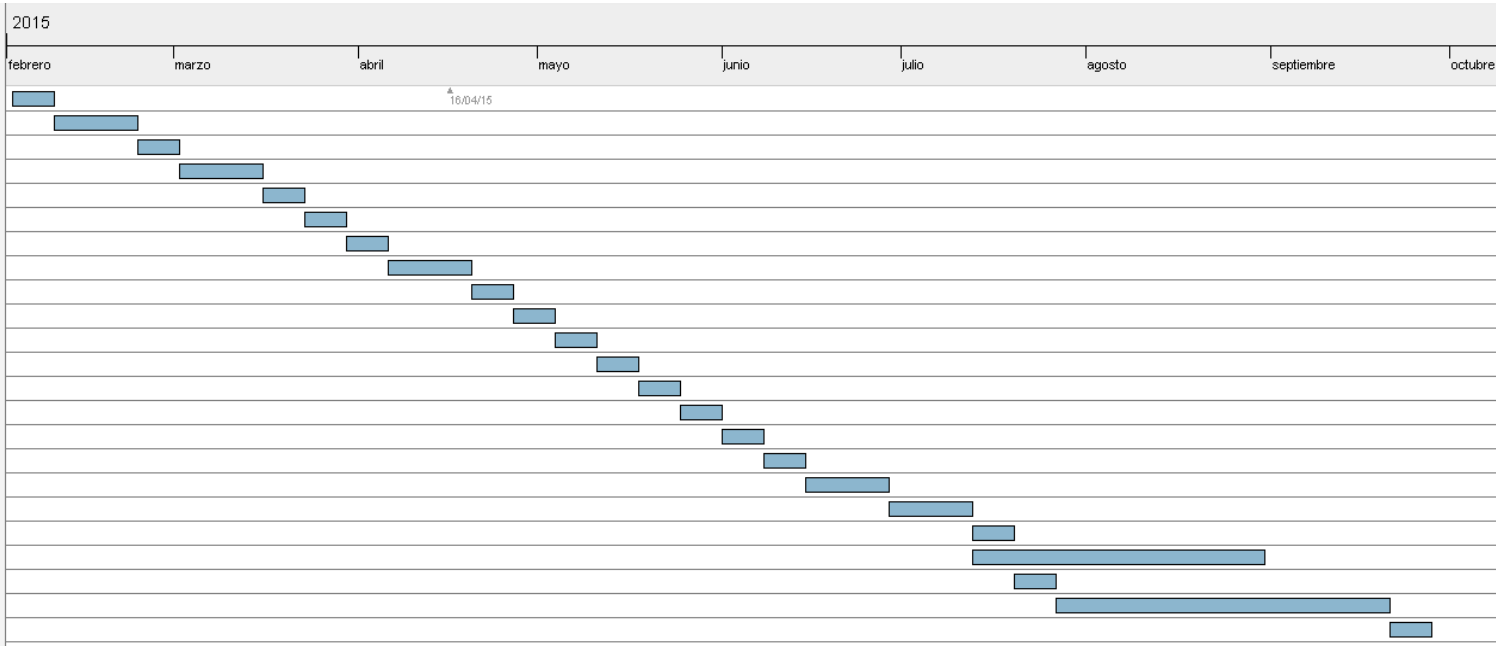

Ilustración 21: Diagrama de Gantt del proyecto.

La planificación de este proyecto se limita al tiempo de dedicación en el que se traducen los créditos universitarios ECTS, correspondientes al Trabajo Fin de Grado. Dado que los créditos del Trabajo Fin de Grado son 12, y el número de horas por crédito se considera como 25 horas, el número total de horas de dedicación debe estar comprendido en las 300 horas de dedicación.

Estas 300 horas, han sido repartidas en el período comprendido entre el mes febrero y el mes septiembre del año 2015. El presupuesto del proyecto se basa en estas 300 horas de dedicación.

5.2. Presupuesto

En esta sección se detalla en presupuesto en personal y materiales asignado a este proyecto. Acorde a las 300 horas de dedicación estimadas previamente, la dedicación en meses se ha calculado de la siguiente manera. Se considera que cada mes tiene 4 semanas, y que en cada semana se realizan 30 horas de dedicación. Esto supone 120 horas de dedicación por mes. Para cumplir con las 300 horas estimadas en la planificación del proyecto, son necesarios 2,5 meses de dedicación.

5.2.1. Personal

En la siguiente tabla se muestra el presupuesto de personal para el desarrollo del proyecto:

Miembro	Salario base mensual	Dedicación (meses)	Coste	Coste con SS*
Programador	1.300,00 €	2	2.600,00 €	3.335,80 €
Analista	1.800,00 €	0,5	900,00 €	1.154,70 €
TOTAL				4.490,50 €

Tabla 84: Presupuesto de personal.

* La cotización de Seguridad Social se calcula como el 28.30% del salario base mensual, por lo que el Coste total con la Seguridad Social se calcula como el Coste total multiplicado por 1.283.

5.2.2. Material

En este apartado se detalla el coste material dedicado al proyecto. Este coste se divide en material hardware y material software. Los precios de todos los materiales incluyen el Impuesto al Valor Agregado I.V.A.

*La amortización de los materiales se calcula como:

$$Amortización = \frac{Dedicación}{Depreciación} \times Precio$$

Ecuación 5: Fórmula de cálculo de la amortización.

5.2.2.1. Hardware

Artículo	Precio	Dedicación (meses)	Período de depreciación (meses)	Amortización (meses)*
Ordenador portátil Mountain Jade 14"	1.200,00 €	2,5	48	62,50 €
Ordenador de sobremesa PcCom Apocalypse	1.300,00 €	2,5	48	67,71 €
TOTAL				130,21 €

Tabla 85: Presupuesto de material hardware.

5.2.2.2. Software

Artículo	Precio	Dedicación (meses)	Período de depreciación (meses)	Amortización (meses)*
Windows 10	0,00 €	2,5	48	0,00 €
Eclipse Luna	0,00 €	2,5	48	0,00 €
Gephi v0.8.2	0,00 €	2,5	0	0,00 €
Microsoft Office 365 Universitarios	80,00 €	2,5	48	4,17 €
TOTAL				4,17 €

Tabla 86: Presupuesto de material software.

5.2.2.3. Total

En la siguiente tabla se muestra el total de coste para materiales, tanto hardware como software:

Material	Precio
Hardware	130,21 €
Software	4,17 €
TOTAL	134,38 €

Tabla 87: Presupuesto total de materiales.

5.2.3. Total

En esta sección se muestra en formato de tabla el desglose total del presupuesto del proyecto:

Concepto	Precio
Personal	4.490,50 €
Material	134,38 €
TOTAL	4.624,88 €

Tabla 88: Presupuesto total del proyecto.

El coste total para la realización del proyecto asciende a CUATRO MIL SEISCIENTOS VEINTICUATRO CON OCHENTA Y OCHO EUROS (4.624,88€).

5.3. Metodología de desarrollo

Para la realización de este proyecto se ha seguido la metodología de Desarrollo Ágil de Software. Esta metodología de realización de proyectos software fundamenta la cooperación directa con el cliente para definir y realizar el proyecto en cuestión. En este caso los clientes de este proyecto han sido los tutores y el desarrollador he sido yo.

Tal y como especifica el Desarrollo Ágil de Software, se debe hacer un seguimiento a corto plazo del desarrollo del proyecto. En este caso, los clientes y yo hemos tenido revisiones periódicas, tanto del desarrollo del software, como de la elaboración de la memoria, en las que hemos definido las diferentes tareas y su planificación dentro de la realización del proyecto. En cada revisión se han expuesto las dudas que han surgido durante las tareas designadas, así como propuestas y modificaciones respecto a los objetivos y el alcance del proyecto.

Cada paso en el desarrollo del proyecto ha sido seguido continuamente y aprobado por los clientes. Todo este proceso ha generado un producto, en este caso, software y memoria acordes a los objetivos y especificaciones de los clientes.

Capítulo 6: Conclusiones y trabajos futuros

6.1. Conclusiones

La realización de este proyecto ha resultado en la implementación de una herramienta capaz de analizar secuencias de eventos independientemente del dominio escogido. El objetivo general se ha visto alcanzado con la implementación de la herramienta.

Respecto a los objetivos específicos impuestos, se han cumplido todos.

Con la implementación de una interfaz gráfica, la herramienta presenta una interfaz de uso cómoda, funcional y amigable. Las métricas con las que la herramienta está dotada permiten modelar un agente en base a la relevancia de sus acciones. Por último, gracias a la exportación del trie en formato gráfico y con la aplicación de colores sobre los nodos y enlaces del trie, el analista es capaz de detectar nodos de interés o secuencias de acciones relevantes a simple vista.

Con la realización de la herramienta, el trabajo del analista encargado de analizar y estudiar secuencias de eventos o acciones se ve enormemente aligerado, ya que el proceso que realiza la herramienta se hacía manualmente antes.

Con todo esto, se da concluido como exitosa la realización de este proyecto.

6.2. Trabajos futuros

Respecto a los trabajos futuros que pueden servir para ampliar la funcionalidad y rendimiento de la herramienta implementada, se han considerado los siguientes:

- Paralelizar el backend del sistema. La paralelización del proceso que calcula todos los parámetros del trie y controla su creación supondría un aumento en el rendimiento de la herramienta. Los tiempos de creación del trie serían menores, por lo que su uso ahorraría todavía más tiempo al analista que utiliza la herramienta.
- Poder cambiar la interfaz de idioma. Dotar la interfaz gráfica de diversos idiomas ayudaría a aquellos que no están familiarizados con el Inglés a utilizar la herramienta.
- Incluir como método de segmentación el criterio de temporalidad entre los eventos. Incluir este método de segmentación de secuencias de eventos ampliaría la funcionalidad de la herramienta. Con éste, se podría realizar un estudio más completo de las secuencias de eventos, ya que la temporalidad entre eventos es un criterio a tener en cuenta en su interpretación.
- Incluir Chi-Cuadrado como métrica. Tal y como se presenta esta métrica de relevancia de eventos en la Tesis Doctoral *Modelado Automático del Comportamiento de Agentes Inteligentes* [62], la inclusión de esta métrica ampliaría el criterio de medición de la relevancia de los eventos en una secuencia de eventos. El modelado de un agente tomaría otro enfoque estadístico.

Bibliografía

[1] H. Kitano, M. Tambe, P. Stone, M. Veloso, S. Coradeschi, E. Osawa, H. Matsubara, I. Noda, y M. Asada. (1997). *The RoboCup Synthetic Agent Challenge 97*. In Proceedings of 15th International Joint Conference on Artificial Intelligence (IJCAI-97), pp. 24–29.

[2] Josep Lluís Cano. (2007). *Business Intelligence: Competir con Información*. pp. 319.

[3] Adam Csapo, Emer Gilmartin, Jonathan Grizou, JingGuang Han, Raveesh Meena, Dimitra Anastasiou, Kristiina Jokinen, and Graham Wilcock. *Multimodal Conversational Interaction with a Humanoid Robot*.

[4] Victor Ng-Thow-Hing, Jongwoo Lim, Joel Wormer, Ravi Kiran Sarvadevabhatla, Carlos Rocha, Kikuo Fujimura, and Yoshiaki Sakagami. *The Memory Game: Creating a human-robot interactive scenario for ASIMO*.

[5] Mirko Suznjevic, Ivana Stupar, Maja Matijasevic. *MMORPG Player Behavior Model based on Player Action Categories*.

[6] Engin Kirda and Christopher Kruegel. *Behavior-based Spyware Detection*.

[7] Schmidt, C., Sridharan, N. S., & Goodson,J. L. (1978). *The Plan Recognition Problem: An Intersection of Psychology and Artificial Intelligence*. Artificial Intelligence, pp. 45-83.

[8] Kautz, H. A., & Allen, J. F. (1986). *Generalized Plan Recognition*. Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI), pp. 32-37.

[9] Charniak, E., & Goldman, R. P. (1993). *A Bayesian Model of Plan Recognition*. Artificial Intelligence, 64, 53-79.

[10] Sidner, C. L., & Israel, D. J. (1981). *Recognizing Intended Meaning and Speakers' Plans*. Proceedings of International Joint Conference on Artificial Intelligence, pp. 203-208. William Kaufmann.

[11] Allen, J. (1983). *Recognizing Intentions from Natural Language Utterances*. In M. Brady, & R. C. Berwick (Eds.), Computational Models of Discourse. Cambridge, Massachusetts: MIT Press.

[12] Zukerman, I., & Albrecht, D. W. (2001). *Predictive Statistical Models for User Modeling*. User Modeling and User-Adapted Interaction, 11 (1), 5-18.

[13] Chen, L., Hoey, J., Nugent, C. D., Cook, D. J., & Yu, Z. (2012). *Sensor-Based Activity Recognition*. IEEE Transactions on Systems, Man, and Cybernetics- Part C: Applications and Reviews, 42 (6), 790-808.

[14] Ke, S.-R., Thuc, H., Lee, Y.-J., Hwang, J.-N., Yoo, J.-H., & Choi, K.-H. (2013). *A Review on Video-Based Human Activity Recognition. Computers*, 2 (2), 88--131.

[15] Chan, M., Estève, D., Escriba, C., & Campo, E. (2008). *A review of smart homes - Present state and future challenges*. Computer Methods and Programs in Biomedicine, 91 (1), 55-81.

[16] Li, N., Cohen, W. W., Koedinger, K. R., & Matsuda, N. (2011). *A Machine Learning Approach for Automatic Student Model Discovery*. Proceedings of the 4th International Conference on Educational Data Mining, pp. 31-40. Eindhoven: www.educationaldatamining.org.

[17] Vail, D. L., & Veloso, M. M. (2008). *Feature Selection for Activity Recognition in Multi-Robot Domains*. Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, pp. 1415-1420. Chicago: AAAI Press.

[18] Ledezma, A., Aler, R., Sanchis, A., & Borrajo, D. (2009). *OMBO: An opponent modeling approach*. AI Communications, 22 (1), 21-35.

[19] Iglesias, J. A., Ledezma, A., & Sanchis, A. (2009). *CAOS Coach 2006 Simulation Team: An opponent modelling approach*. Computing and Informatics Journal, 28, 57-80.

[20] Salah, A. A., del Solar Ruiz, J., Meriçli, Ç., & Oudeyer, P.-Y. (2012). *Human Behavior Understanding for Robotics*. Human Behavior Understanding - Third International Workshop, HBU 2012, pp. 1-16. Vilamoura: Springer.

[21] Bao, L., & Intille, S. S. (2004). *Activity recognition from user-annotated acceleration data*. Pervasive Computing. 3001, pp. 1-17. Springer-Verlag.

[22] Stikic, M., & Schiele, B. (2009). *Activity Recognition from Sparsely Labeled Data Using Multi-Instance Learning*. International Symposium on Location and Context Awareness (LoCA), 5561, pp. 156-173.

[23] Elangovan, V., Bandaru, V. K., & Shirkhodaie, A. (2012). *Team activity analysis and recognition based on Kinect depth map and optical imagery techniques*. Signal Processing, Sensor Fusion, and Target Recognition XXI. Baltimore.

[24] Kriegel, H.-P., Kröger, P., & Zimek, A. (2010). *Outlier Detection Techniques*. Tutorial at 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD). Washington, D.C.

[25] Palaniappan, A., Bhargavi, R., & Vaidehi, V. (2012). *Abnormal human activity recognition using SVM based approach*. 2012 International Conference on Recent Trends In Information Technology (ICRTIT), (pp. 97-102).

[26] Munguia Tapia, E., Intille, S. S., & Larson, K. (2004). *Activity Recognition in the Home Using Simple and Ubiquitous Sensors*. Pervasive Computing, Second International Conference, PERVASIVE 2004, pp. 158-17. Vienna: Springer

[27] Srivastava, M., Muntz, R., & Potkonjak, M. (2001). *Smart Kindergarten: Sensor-based Wireless Networks for Smart Developmental Problem-solving Environments*. 7th Annual International Conference on Mobile Computing and Networking, pp. 132-138. Rome: ACM.

[28] Iglesias, J. A., Angelov, P., Ledezma, A., & Sanchis, A. (2010). *Human Activity Recognition Based on Evolving Fuzzy Systems*. International Journal of Neural Systems, 20 (5), 355-364.

[29] Martin Gayral, B. (2007). *Diseño y desarrollo de un jugador inteligente para la competición anual de agentes de póquer*.

[30] Carbó Rubiera, J., & Ledezma, A. (2003). *A Machine Learning Based Evaluation of a Negotiation between Agents Involving Fuzzy Counter-Offers*. Web Intelligence, First International Atlantic Web Intelligence Conference, AWIC 2003, pp. 268-277. Madrid: Springer.

[31] Duong, T. V., Bui, H. H., Phung, D. Q., & Venkatesh, S. (2005). *Activity recognition and abnormality detection with the switching hidden semi-Markov model*. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 838-845. IEEE Computer Society.

[32] Kuo, Y.-M., Lee, J.-S., & Chung, P.-C. (2010). *A Visual Context-Awareness-Based Sleeping-Respiration Measurement System*. IEEE Transactions on Information Technology in Biomedicine, 14 (4), 255-265.

[33] Chan, M., Estève, D., Escriba, C., & Campo, E. (2008). *A review of smart homes - Present state and future challenges*. Computer Methods and Programs in Biomedicine, 91 (1), 55-81.

[34] Nugent, C., Mulvenna, M., Hong, X., & Devlin, S. (2009). *Experiences in the development of a Smart Lab*. International Journal of Biomedical Engineering and Technology, 2 (4), 319-331.

[35] Wojek, C., Nickel, K., & Stiefelhagen, R. (2006). *Activity Recognition and Room-Level Tracking in an Office Environment*. 2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 25-30.

[36] Ohn-Bar, E., & Trivedi, M. (2013). *In-vehicle hand activity recognition using integration of regions*. 2013 IEEE Intelligent Vehicles Symposium (IV), pp. 1034-1039. IEEE.

[37] Veeraraghavan, H., Bird, N., Atev, S., & Papanikolopoulos, N. (2007). *Classifiers for Driver Activity Monitoring*. Transportation Research Part C, 15 (1), 51-67.

[38] Ke, Y., Sukthankar, R., & Hebert, M. (2007). *Spatio-temporal Shape and Flow Correlation for Action Recognition*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-8. Minneapolis, MN: IEEE.

[39] Bodor, R., Jackson, B., & Papanikolopoulos, N. (2003). *Vision-based Human Tracking and Activity Recognition*. 11th Mediterranean Conference on Control and Automation, 1, pp. 18-20. Rhodes, Greece.

[40] Huo, F., Hendriks, E., Paclik, P., & Oomes, A. (2009). *Markerless Human Motion Capture and Pose Recognition*. 10th IEEE Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), pp. 13-16. London, UK.

[41] Wren, C., Azarbayejani, A., Darrell, T., & Pentland, A. (1997). *Pfinder: Real-time tracking of the human body*. IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 780-785.

[42] Sempena, S., Maulidevi, N., & P.R., A. (2011). *Human Action Recognition Using Dynamic Time Warping*. IEEE International Conference on Electrical Engineering and Informatics (ICEEI), pp. 1-5. Bandung, Indonesia: IEEE.

[43] Ribeiro, P., & Santos-Victor, J. (2005). *Human Activity Recognition from Video: Modeling, Feature Selection and Classification Architecture*. International Workshop on Human Activity Recognition and Modelling (HAREM), 1, pp. 61-70.

[44] Niu, W., Long, J., Han, D., & Wang, Y. (2004). *Human Activity Detection and Recognition for Video Surveillance*. IEEE International Conference on Multimedia and Expo (ICME), pp. 719-722. Taipei, Taiwan: IEEE.

[45] Geib, C. W., & Goldman, R. P. (2001). *Plan Recognition in Intrusion Detection Systems*. DARPA Information Survivability Conference and Exposition (DISCEX).

[46] Jecheva, V. (2006). *About Some Applications of Hidden Markov Model in Intrusion Detection Systems*. International Conference on Computer Systems and Technologies - CompSys - Tech'06.

[47] Kichkaylo, T., Ryutov, T., Orosz, M. D., & Neches, R. (2010). *Planning to Discover and Counteract Attacks*. Informatica (Slovenia), 34 (2), pp. 159-168.

[48] Kenyeres, P., Szentgyorgyi, A., Meszaros, T., & Feher, G. (2010). *BotSpot: Anonymous and Distributed Malware Detection*. The Second International Conference on Wireless & Mobile Networks (WiMo-2010).

[49] Filiol, E., & Josse, S. (2012). *New Trends in Security Evaluation of Bayesian Network-Based Malware Detection Models*. 45th Hawaii International Conference on System Science (HICSS), pp. 5574 - 5583.

[50] Sung, M., DeVaul, R. W., Jimenez, S., Gips, J., & Pentland, A. (2004). *Shiver Motion and Core Body Temperature Classification for Wearable Soldier Health Monitoring Systems*. Eighth International Symposium on Wearable Computers (2004), pp. 192-193. IEEE Computer Society.

[51] Sukthankar, G. (Julio de 2007). *Activity Recognition for Agent Teams*.

[52] Tambe, M., & Rosenbloom, P. S. (1995). *RESC: An Approach for Real-time, Dynamic Agent Tracking*. Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, pp. 103-111. Montréal Québec.

[53] Carberry, S. (2001). *Techniques for plan recognition*. User Modeling and User-Adapted Interaction, 11 (1-2), pp. 31-48.

[54] Webb, G. I., Pazzani, V., & Billsus, V. (2001). *Machine learning for user modeling*. User Modeling and User-Adapted Interaction, 11, pp. 19-20.

[55] T. S. Tan, F. Dillon, E. Hadzic, and E. Chang. (2006). *In SEQUEST: mining frequent subsequences using DMA Strips*. Proceedings of the 7th International Conference on Data Mining and Information Engineering, pp. 35-328.

[56] P. Laird and R. Saul. (1994). *Discrete sequence prediction and its applications*. Machine Learning, 15(1), 43–68.

[57] M. Bicego, V. Murino, and M. A.T. Figueiredo. (2004). *Similarity-based classification of sequences using hidden markov models*. Pattern Recognition, 37(12), 2281 – 2291.

[58] J. Yang and W. Wang. (2003). *Cluseq: Efficient and effective sequence clustering*. In Proceedings of the International Conference on Data Engineering (ICDE-03), pp. 101–112. IEEE Press.

[59] Q. Ma, J. T. Wang, D. Shasha, and C. H. Wu. (2001). *DNA sequence classification via an expectation maximization algorithm and neural networks: a case study*. IEEE Transactions on Systems, Man, and Cybernetics, Part C, 31(4), 468–475.

[60] S. E. Coull, J. W. Branch, B. K. Szymanski, and E. Breimer. (2003). *Intrusion detection: A bioinformatics approach*. In Proceedings of the Annual Computer Security Applications Conference (ACSAC-03), pp. 24–33.

[61] G. Kaminka, M. Fidanboylu, A. Chang, and M. Veloso. (2002). *Learning the sequential coordinated behavior of teams from observations*. In Proceedings of the Robot Soccer World Cup VI (RoboCup-02).

[62] Iglesias, J. A. (2010). *Tesis doctoral: Modelado Automático del Comportamiento de Agentes Inteligentes*.

[63] The RoboCup Coach Competition Web Page.

[64] C. Basu, H. Hirsh, and W. W. Cohen. (1998). *Recommendation as classification: Using social and content-based information in recommendation*. In Proceedings of the National Conference on Artificial Intelligence(AAAI-98), pp. 714–720.

[65] Fredkin, E. (1960). *Trie Memory*. Communication of the ACM. Vol. 3:9. pp. 490-499.

[66] Disponible en: http://linux.thai.net/~thep/datrie/datrie.html

[67] Disponible en: https://www.java.com/en/about/

[68] Java Fundamentals. Disponible en: http://www.oracle.com/events/global/en/java-outreach/resources/java-a-beginners-guide-1720064.pdf

[69] Disponible en: http://www.uml.org/

[70] Disponible en: https://eclipse.org/org/

[71] Cultoromics 2.0: Forecasting large-scale human behaviour using global news media tone in time and space. Disponible en: http://journals.uic.edu/ojs/index.php/fm/article/view/3663/3040

[72] Collateral Damage: #Oslo Attacks and Proliferating Islamophobia. Disponible en: http://www.jadaliyya.com/pages/index/2343/collateral-damage_#oslo-attacks-and-proliferating-

[73] Disponible en: http://gephi.github.io/about/

[74] Disponible en: http://staruml.io/

[75] M. Schonlau, W. Dumouchel, W. H. Ju, A. F. Karr, and Theus. (2001). *Computer intrusion: Detecting masquerades*. In Statistical Science, 16, pp 58–74.

[76] M. Schonlau. (2009). *Matt Schonlau web page - masquerading user data*. Disponible en: http://www.schonlau.net/intrusion.html

Anexo I: Resultados de la experimentación

Usuario 2

Profundidad 3	
Valor de soporte	Nodo
0,138	Root-gcc
0,091	Root-awk
0,081	Root-less
0,07	Root-nawk
0,069	Root-uname
0,069	Root-uname-nawk
0,069	Root-gcc-gcc
0,057	Root-awk-less
0,053	Root-uname-nawk-cpp
0,053	Root-nawk-cpp
0,053	Root-cpp
0,052	Root-nawk-cpp-cc1
0,052	Root-cpp-cc1
0,052	Root-cc1
0,044	Root-cpp-cc1-as
0,044	Root-cc1-as
0,044	Root-as
0,043	Root-make
0,042	Root-cat
0,036	Root-less-awk

Tabla 89: Tabla de valores de soporte para Usuario 2 con Profundidad 3.

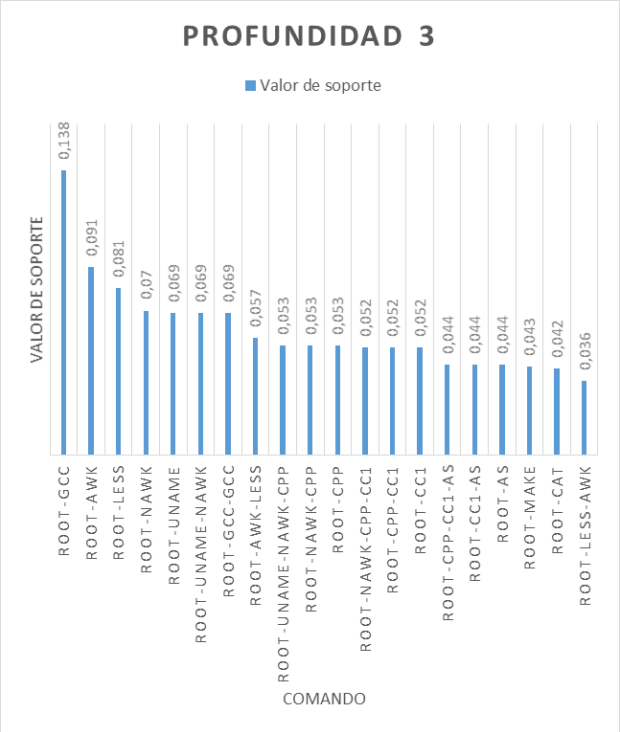

Ilustración 22: Gráfica de valores de soporte para Usuario 2 con Profundidad 3.

Profundidad 5	
Valor de soporte	Nodo
0,138	Root-gcc
0,091	Root-awk
0,081	Root-less
0,07	Root-nawk
0,069	Root-uname
0,069	Root-uname-nawk
0,069	Root-gcc-gcc
0,057	Root-awk-less
0,053	Root-uname-nawk-cpp
0,053	Root-nawk-cpp
0,053	Root-cpp
0,052	Root-uname-nawk-cpp-cc1
0,052	Root-nawk-cpp-cc1
0,052	Root-cpp-cc1
0,052	Root-cc1
0,044	Root-uname-nawk-cpp-cc1-as
0,044	Root-nawk-cpp-cc1-as
0,044	Root-cpp-cc1-as
0,044	Root-cc1-as
0,044	Root-as

Tabla 90: Tabla de valores de soporte para Usuario 2 con Profundidad 5.

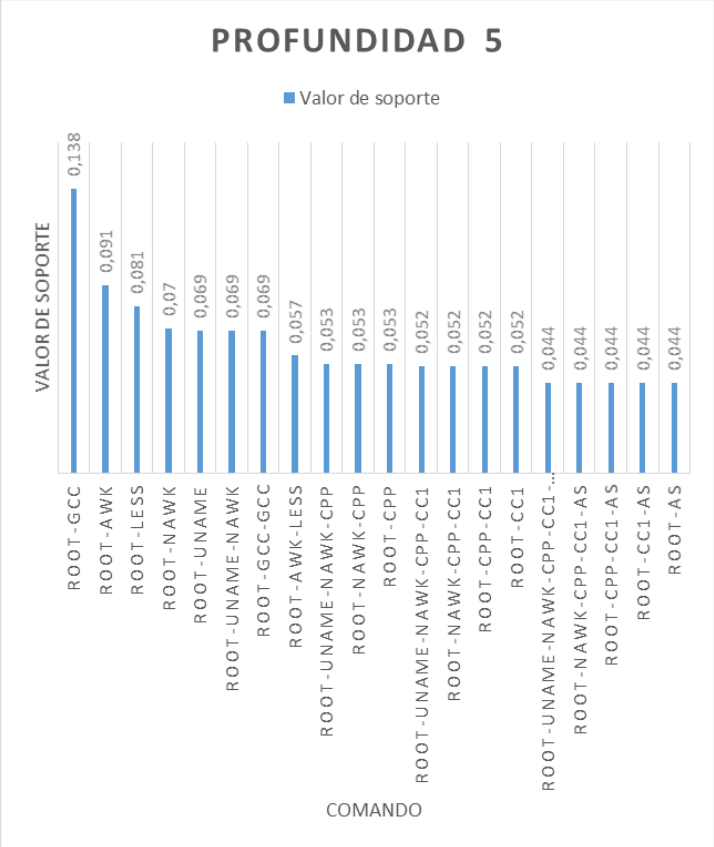

Ilustración 23: Gráfica de valores de soporte para Usuario 2 con Profundidad 5.

Profundidad 7	
Valor de soporte	Nodo
0,138	Root-gcc
0,091	Root-awk
0,081	Root-less
0,07	Root-nawk
0,069	Root-uname
0,069	Root-uname-nawk
0,069	Root-gcc-gcc
0,057	Root-awk-less
0,053	Root-uname-nawk-cpp
0,053	Root-nawk-cpp
0,053	Root-cpp
0,052	Root-uname-nawk-cpp-cc1
0,052	Root-nawk-cpp-cc1
0,052	Root-cpp-cc1
0,052	Root-cc1
0,044	Root-uname-nawk-cpp-cc1-as
0,044	Root-nawk-cpp-cc1-as
0,044	Root-cpp-cc1-as
0,044	Root-cc1-as
0,044	Root-as

Tabla 91: Tabla de valores de soporte para Usuario 2 con Profundidad 7.

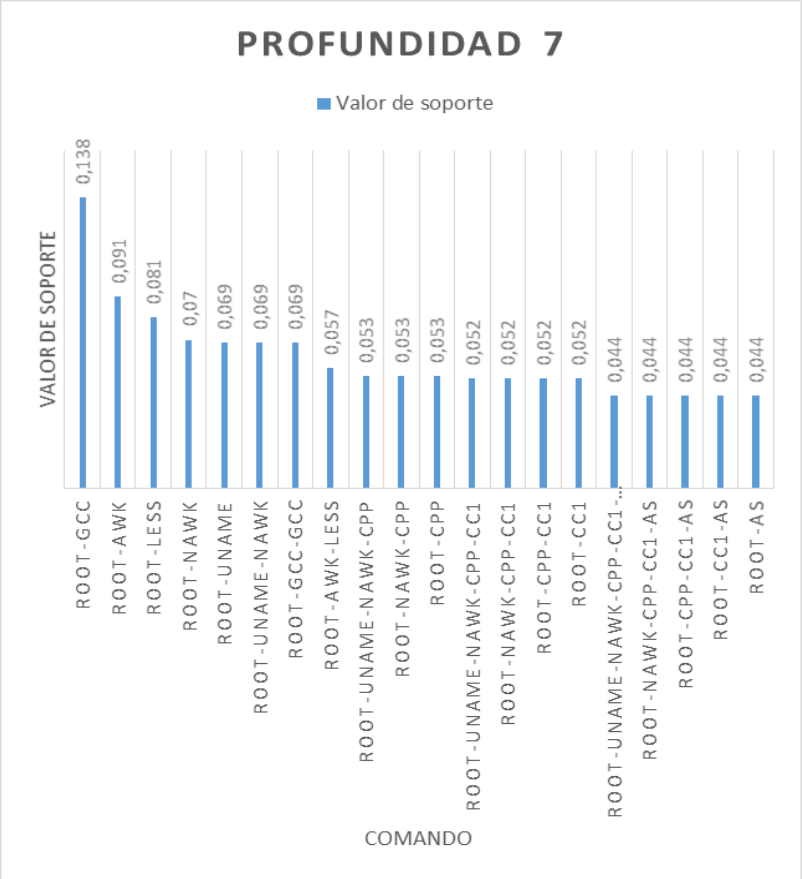

Ilustración 24: Gráfica de valores de soporte para Usuario 2 con Profundidad 7.

Usuario 3

Profundidad 3	
Valor de soporte	Nodo
0,174	Root-egrep
0,154	Root-expr
0,13	Root-egrep-egrep
0,092	Root-expr-expr
0,087	Root-egrep-egrep-egrep
0,087	Root-java
0,061	Root-expr-expr-dirname
0,061	Root-expr-dirname
0,061	Root-dirname
0,044	Root-basename
0,043	Root-.java_wr
0,043	Root-.java_wr-expr
0,043	Root-.java_wr-expr-expr
0,043	Root-expr-dirname-basename
0,043	Root-dirname-basename
0,043	Root-dirname-basename-egrep
0,043	Root-basename-egrep
0,043	Root-basename-egrep-egrep
0,042	Root-java-java
0,036	Root-make

Tabla 92: Tabla de valores de soporte para Usuario 3 con Profundidad 3.

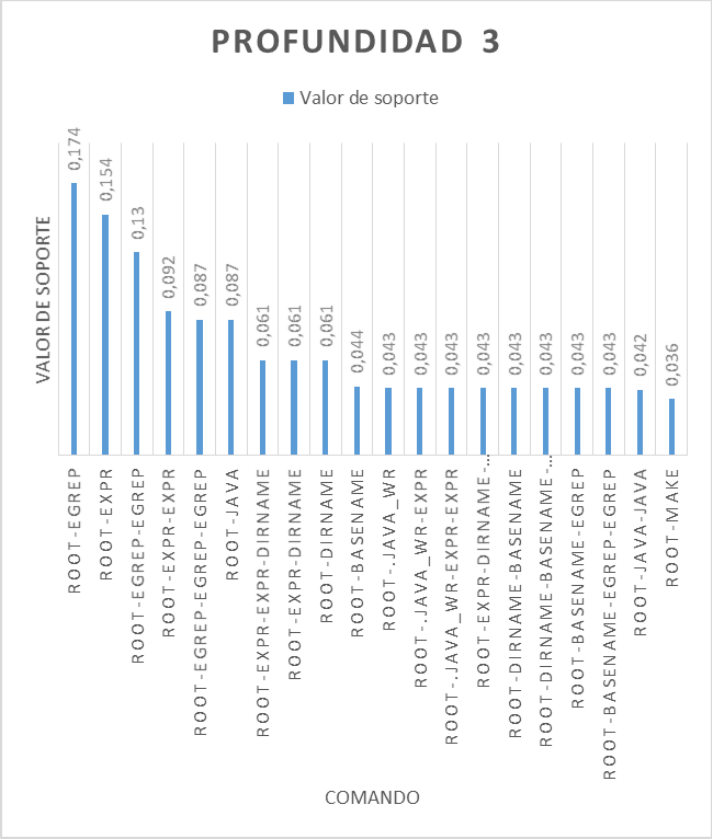

Ilustración 25: Gráfica de valores de soporte para Usuario 3 con Profundidad 3.

Profundidad 5	
Valor de soporte	Nodo
0,174	Root-egrep
0,154	Root-expr
0,13	Root-egrep-egrep
0,092	Root-expr-expr
0,087	Root-egrep-egrep-egrep
0,087	Root-java
0,061	Root-expr-expr-dirname
0,061	Root-expr-dirname
0,061	Root-dirname
0,044	Root-basename
0,043	Root-.java_wr
0,043	Root-.java_wr-expr
0,043	Root-.java_wr-expr-expr
0,043	Root-.java_wr-expr-expr-dirname
0,043	Root-.java_wr-expr-expr-dirname-basename
0,043	Root-expr-expr-dirname-basename
0,043	Root-expr-dirname-basename
0,043	Root-dirname-basename
0,043	Root-expr-expr-dirname-basename-egrep
0,043	Root-expr-dirname-basename-egrep

Tabla 93: Tabla de valores de soporte para Usuario 3 con Profundidad 5.

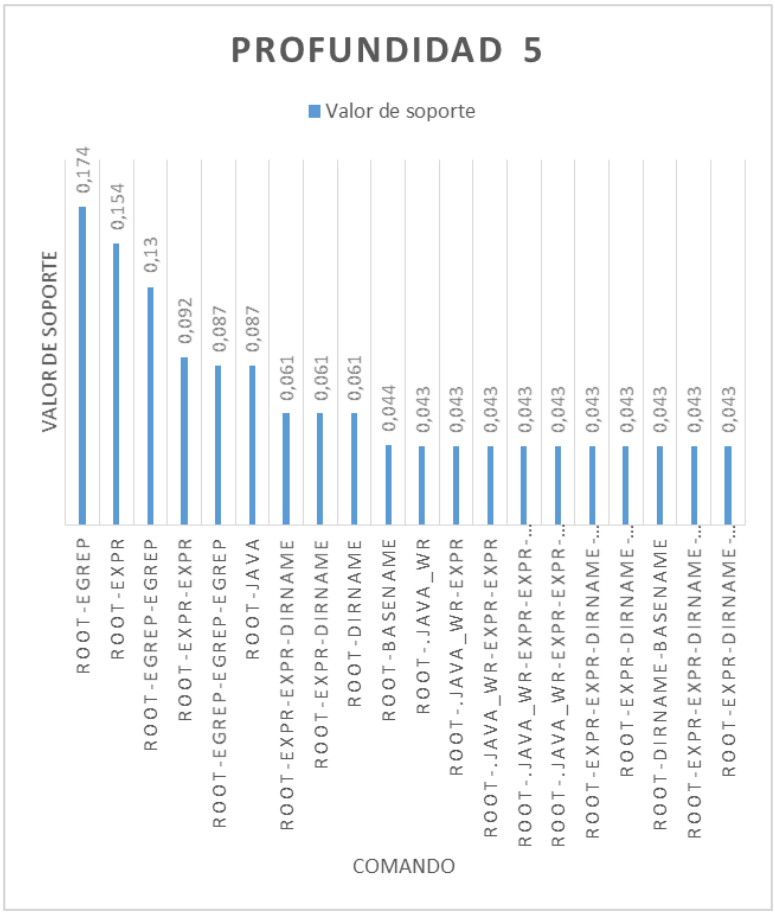

Ilustración 26: Gráfica de valores de soporte para Usuario 3 con Profundidad 5.

Profundidad 7	
Valor de soporte	Nodo
0,174	Root-egrep
0,154	Root-expr
0,13	Root-egrep-egrep
0,092	Root-expr-expr
0,087	Root-egrep-egrep-egrep
0,087	Root-java
0,061	Root-expr-expr-dirname
0,061	Root-expr-dirname
0,061	Root-dirname
0,044	Root-basename
0,043	Root-.java_wr
0,043	Root-.java_wr-expr
0,043	Root-.java_wr-expr-expr
0,043	Root-.java_wr-expr-expr-dirname
0,043	Root-.java_wr-expr-expr-dirname-basename
0,043	Root-expr-expr-dirname-basename
0,043	Root-expr-dirname-basename
0,043	Root-dirname-basename
0,043	Root-.java_wr-expr-expr-dirname-basename-egrep
0,043	Root-expr-expr-dirname-basename-egrep

Tabla 94: Tabla de valores de soporte para Usuario 3 con Profundidad 7.

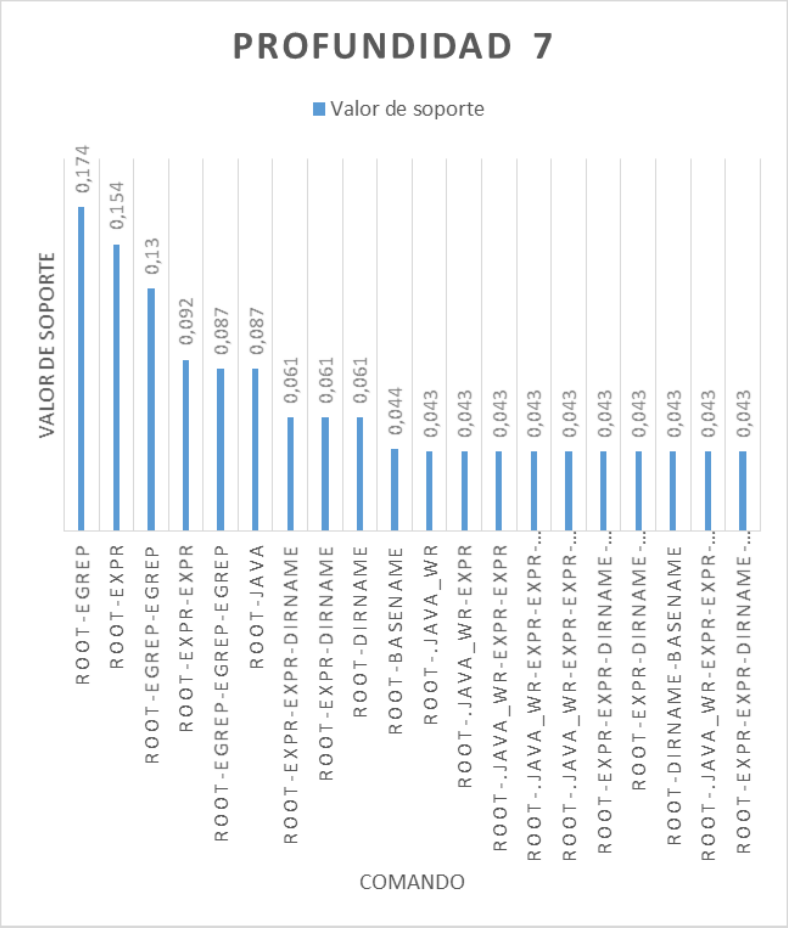

Ilustración 27: Gráfica de valores de soporte para Usuario 3 con Profundidad 7.

Usuario 4

Profundidad 3	
Valor de soporte	Nodo
0,204	Root-sh
0,105	Root-more
0,099	Root-more-sh
0,089	Root-sh-more
0,089	Root-sh-more-sh
0,074	Root-more-sh-more
0,062	Root-csh
0,048	Root-generic
0,042	Root-sendmail
0,041	Root-cat
0,039	Root-MediaMai
0,034	Root-rm
0,026	Root-ls
0,023	Root-toolches
0,023	Root-sh-MediaMai
0,02	Root-cat-mail
0,02	Root-cat-mail-csh
0,02	Root-mail
0,02	Root-mail-csh
0,019	Root-date

Tabla 95: Tabla de valores de soporte para Usuario 4 con Profundidad 3.

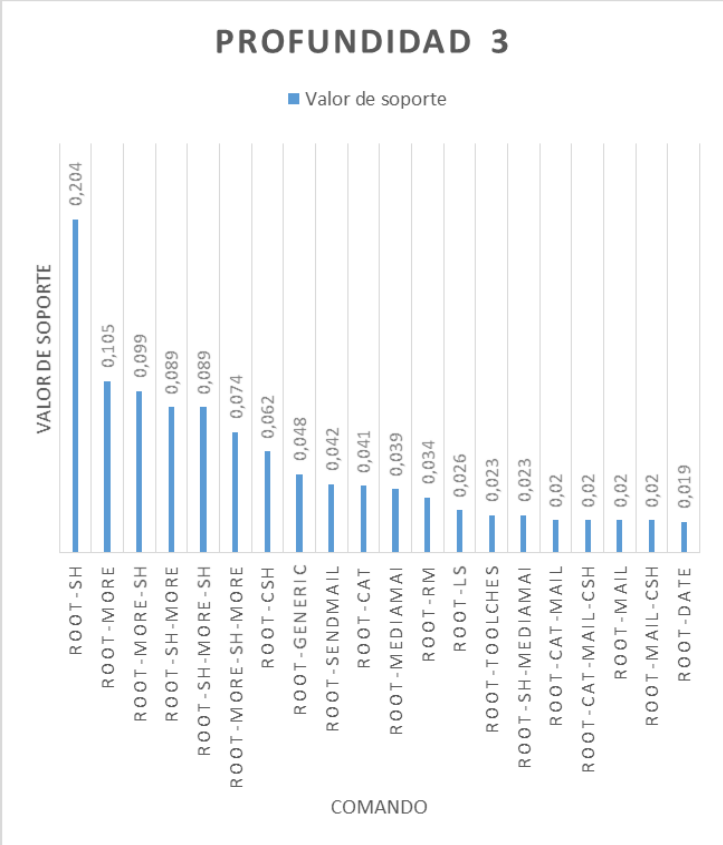

Ilustración 28: Gráfica de valores de soporte para Usuario 4 con Profundidad 3.

Profundidad 5	
Valor de soporte	Nodo
0,204	Root-sh
0,105	Root-more
0,099	Root-more-sh
0,089	Root-sh-more
0,089	Root-sh-more-sh
0,074	Root-more-sh-more
0,074	Root-more-sh-more-sh
0,069	Root-sh-more-sh-more
0,069	Root-sh-more-sh-more-sh
0,062	Root-csh
0,059	Root-more-sh-more-sh-more
0,047	Root-generic
0,042	Root-sendmail
0,041	Root-cat
0,039	Root-MediaMai
0,034	Root-rm
0,026	Root-ls
0,023	Root-toolches
0,023	Root-sh-MediaMai
0,02	Root-cat-mail

Tabla 96: Tabla de valores de soporte para Usuario 4 con Profundidad 5.

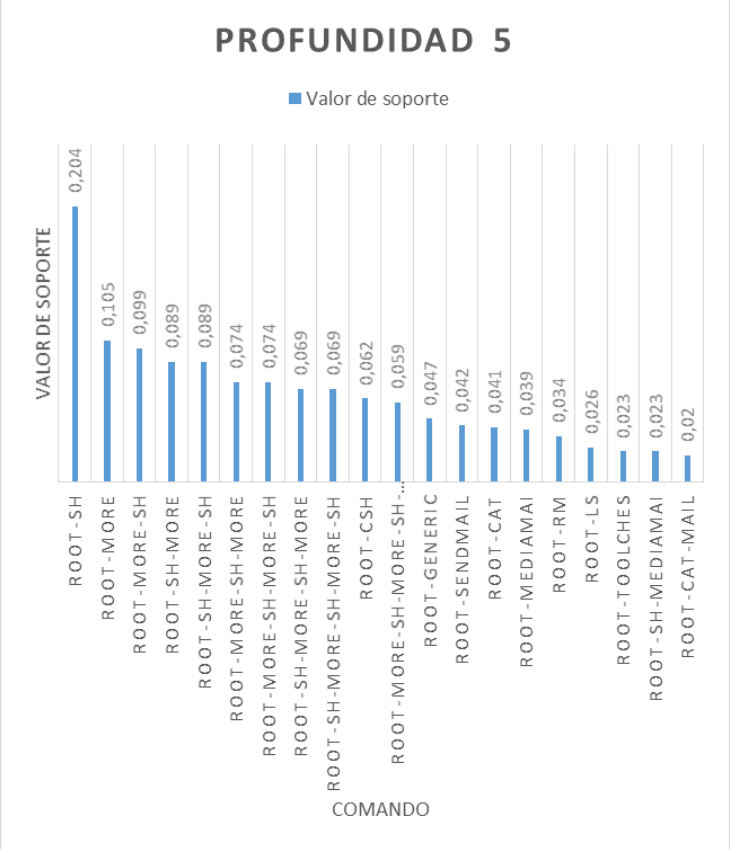

Ilustración 29: Gráfica de valores de soporte para Usuario 4 con Profundidad 5.

Profundidad 7	
Valor de soporte	Nodo
0,204	Root-sh
0,105	Root-more
0,099	Root-more-sh
0,089	Root-sh-more
0,089	Root-sh-more-sh
0,074	Root-more-sh-more
0,074	Root-more-sh-more-sh
0,069	Root-sh-more-sh-more
0,069	Root-sh-more-sh-more-sh
0,062	Root-csh
0,059	Root-more-sh-more-sh-more
0,059	Root-more-sh-more-sh-more-sh
0,056	Root-sh-more-sh-more-sh-more
0,056	Root-sh-more-sh-more-sh-more-sh
0,048	Root-more-sh-more-sh-more-sh-more
0,047	Root-generic
0,042	Root-sendmail
0,041	Root-cat
0,039	Root-MediaMai
0,034	Root-rm

Tabla 97: Tabla de valores de soporte para Usuario 4 con Profundidad 7.

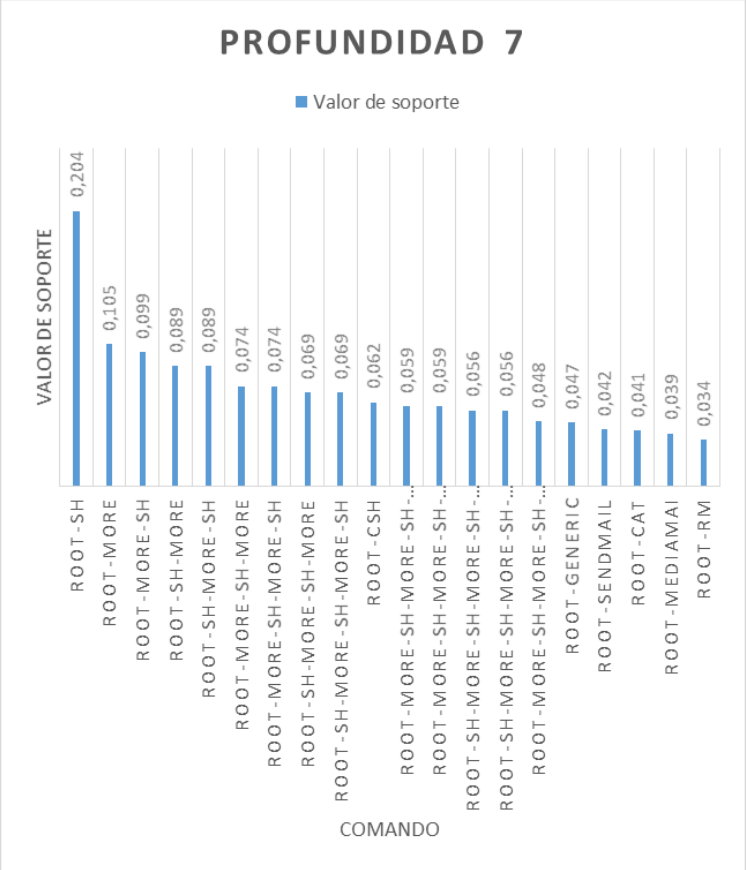

Ilustración 30: Gráfica de valores de soporte para Usuario 4 con Profundidad 7.

Usuario 5

Profundidad 3	
Valor de soporte	Nodo
0,112	Root-generic
0,07	Root-sh
0,05	Root-date
0,049	Root-cat
0,044	Root-true
0,04	Root-tcpostio
0,035	Root-ls
0,034	Root-rm
0,032	Root-date-generic
0,03	Root-ln
0,03	Root-tcpostio-tcpostio
0,027	Root-sed
0,024	Root-p
0,024	Root-p-sh
0,024	Root-lp
0,022	Root-LOCK
0,022	Root-ls-sed
0,022	Root-ls-sed-FIFO
0,022	Root-sed-FIFO
0,022	Root-FIFO

Tabla 98: Tabla de valores de soporte para Usuario 5 con Profundidad 3.

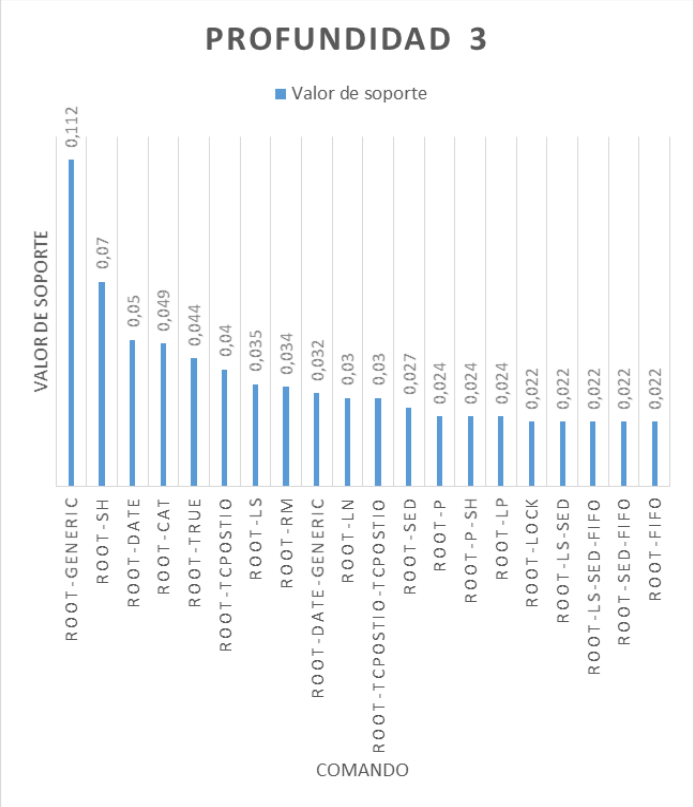

Ilustración 31: Gráfica de valores de soporte para Usuario 5 con Profundidad 3.

Profundidad 5	
Valor de soporte	Nodo
0,112	Root-generic
0,07	Root-sh
0,05	Root-date
0,049	Root-cat
0,044	Root-true
0,04	Root-tcpostio
0,035	Root-ls
0,034	Root-rm
0,032	Root-date-generic
0,03	Root-ln
0,03	Root-tcpostio-tcpostio
0,027	Root-sed
0,024	Root-p
0,024	Root-p-sh
0,024	Root-lp
0,022	Root-LOCK
0,022	Root-ls-sed
0,022	Root-ls-sed-FIFO
0,022	Root-sed-FIFO
0,022	Root-FIFO

Tabla 99: Tabla de valores de soporte para Usuario 5 con Profundidad 5.

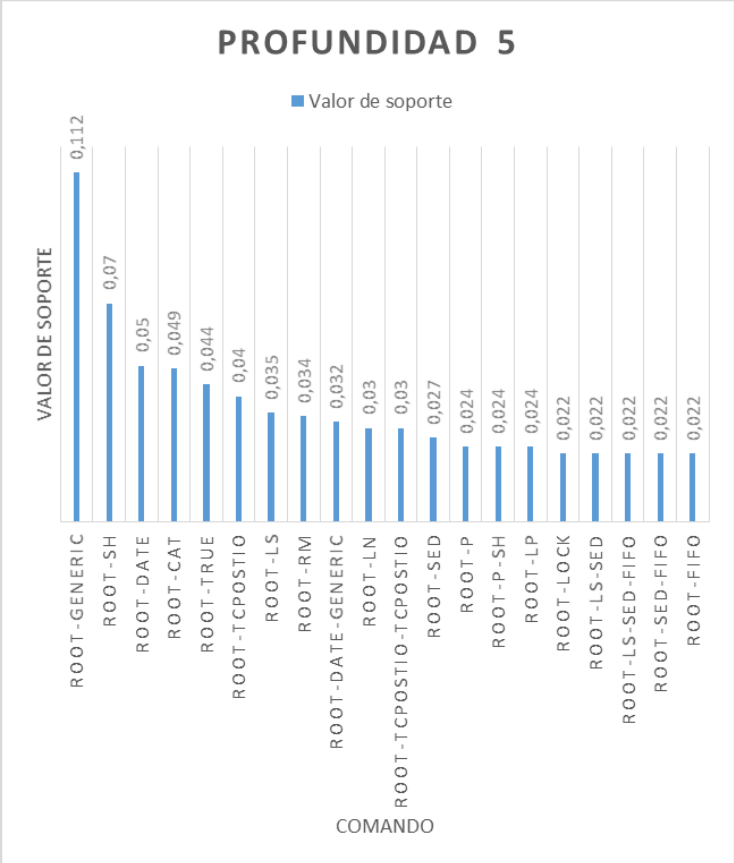

Ilustración 32: Gráfica de valores de soporte para Usuario 5 con Profundidad 5.

Profundidad 7	
Valor de soporte	Nodo
0,112	Root-generic
0,07	Root-sh
0,05	Root-date
0,049	Root-cat
0,044	Root-true
0,04	Root-tcpostio
0,035	Root-ls
0,034	Root-rm
0,032	Root-date-generic
0,03	Root-ln
0,03	Root-tcpostio-tcpostio
0,027	Root-sed
0,024	Root-p
0,024	Root-p-sh
0,024	Root-lp
0,022	Root-LOCK
0,022	Root-ls-sed
0,022	Root-ls-sed-FIFO
0,022	Root-sed-FIFO
0,022	Root-FIFO

Tabla 100: Tabla de valores de soporte para Usuario 5 con Profundidad 7.

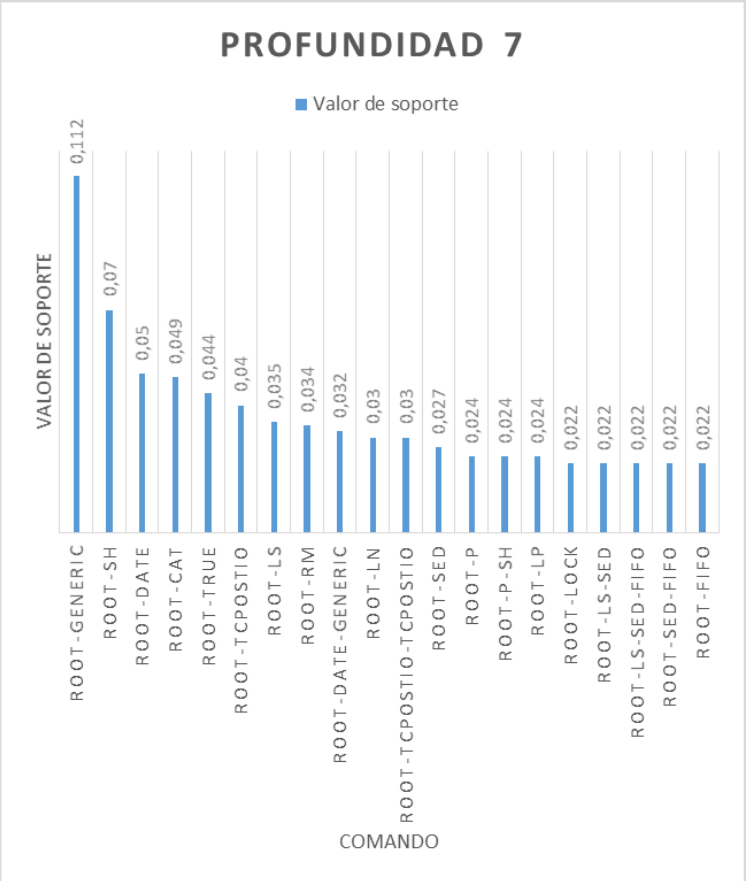

Ilustración 33: Gráfica de valores de soporte para Usuario 5 con Profundidad 7.

Anexo II: Manual de Usuario

En este anexo se desarrolla el Manual de Usuario en el que se explica la estructura y las funcionalidades de la herramienta desarrollada en este proyecto.

1. Estructura de la herramienta

En esta sección del Manual de Usuario se detallan los componentes y funcionalidades de la herramienta desarrollada en este proyecto.

En primer lugar se muestra la interfaz gráfica que posee la herramienta, junto con la identificación de todos sus componentes.

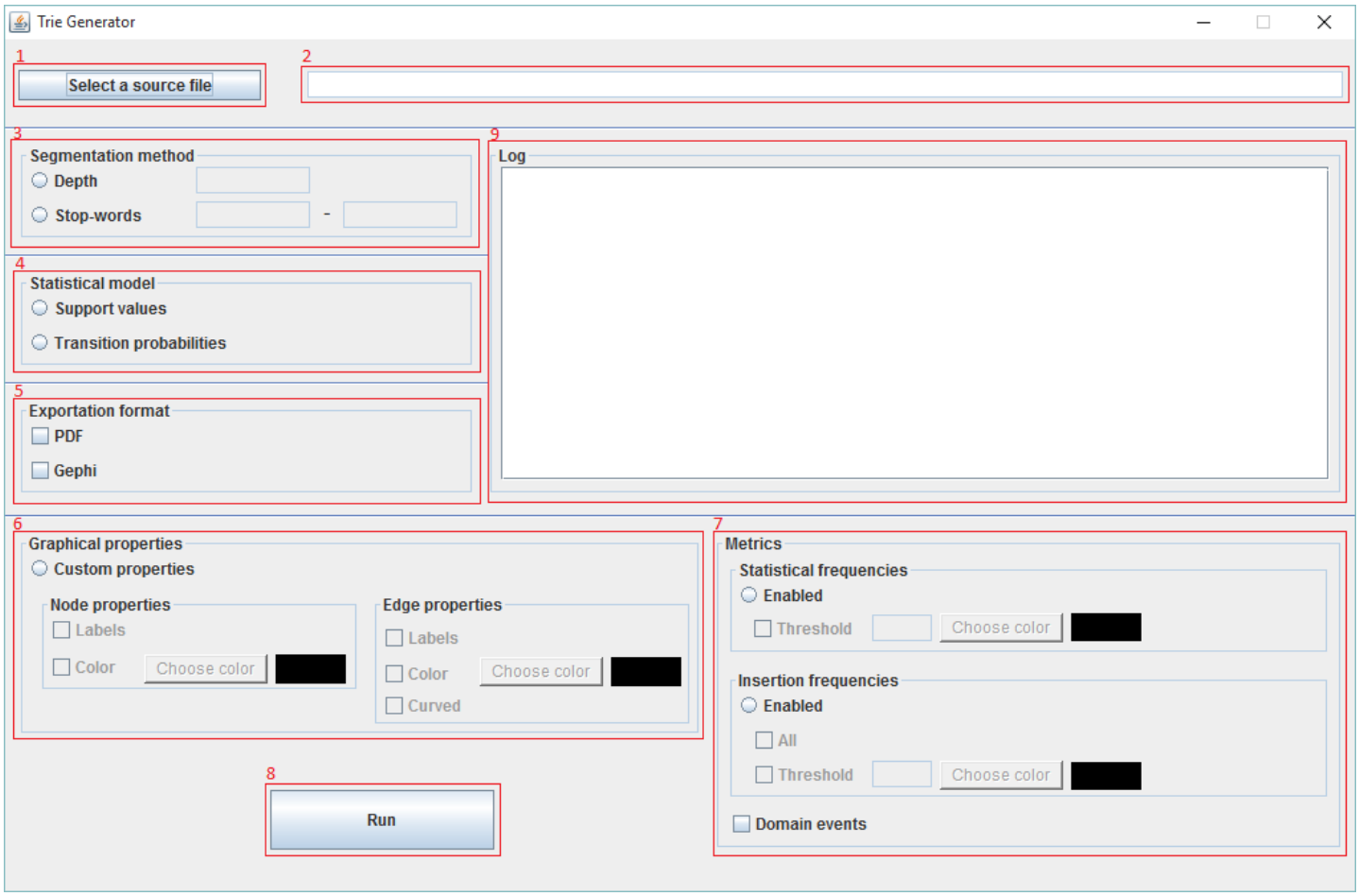

Ilustración 34: Manual de Usuario: estructura de la herramienta.

A continuación se detalla cada componente marcado en la ilustración anterior:

1. Botón de selección de fichero.
2. Ruta del fichero.
3. Panel de Método de Segmentación.
4. Panel de Modelo Estadístico.
5. Panel de Formato de Exportación.
6. Panel de Propiedades Gráficas.

7. Panel de Métricas.
8. Botón para ejecutar el programa.
9. Campo de texto no editable que muestra la ejecución del programa.

Una vez determinados los componentes del programa, se procede a detallarlos individualmente.

1.1. Botón de selección de fichero

Cuando este botón es pulsado se despliega un selector de ficheros, que por defecto tiene establecido la visualización de todos los ficheros cuya extensión es *.txt*. A continuación se muestra este selector:

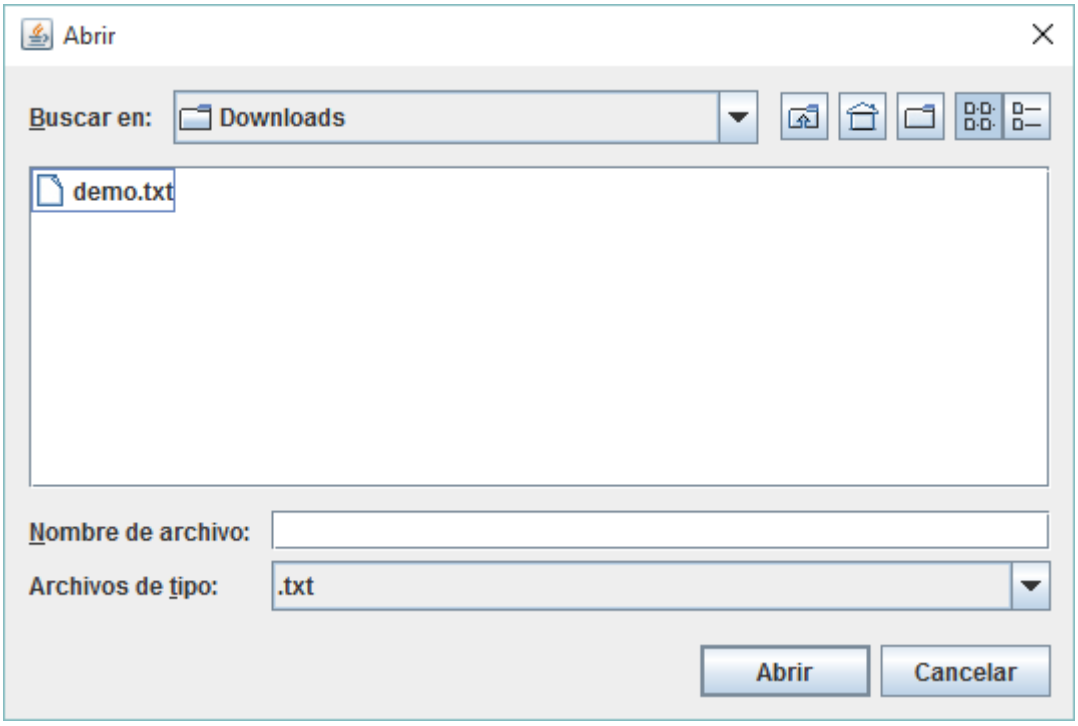

Ilustración 35: Manual de Usuario: botón de selección de fichero.

Para seleccionar un fichero basta con hacer doble click sobre éste, o seleccionar el fichero y pulsar el botón *Abrir*, situado en la parte inferior derecha. Hecho esto, esta ventana se cerrará automáticamente.

En caso de cancelar este proceso, se debe pulsar el botón *Cancelar*, situado en la parte inferior derecha.

1.2. Ruta del fichero

Este campo de texto no es editable y su propósito es mostrar la ruta absoluta que tiene el fichero seleccionado dentro del sistema. En la siguiente ilustración se muestra este campo con un fichero seleccionado:

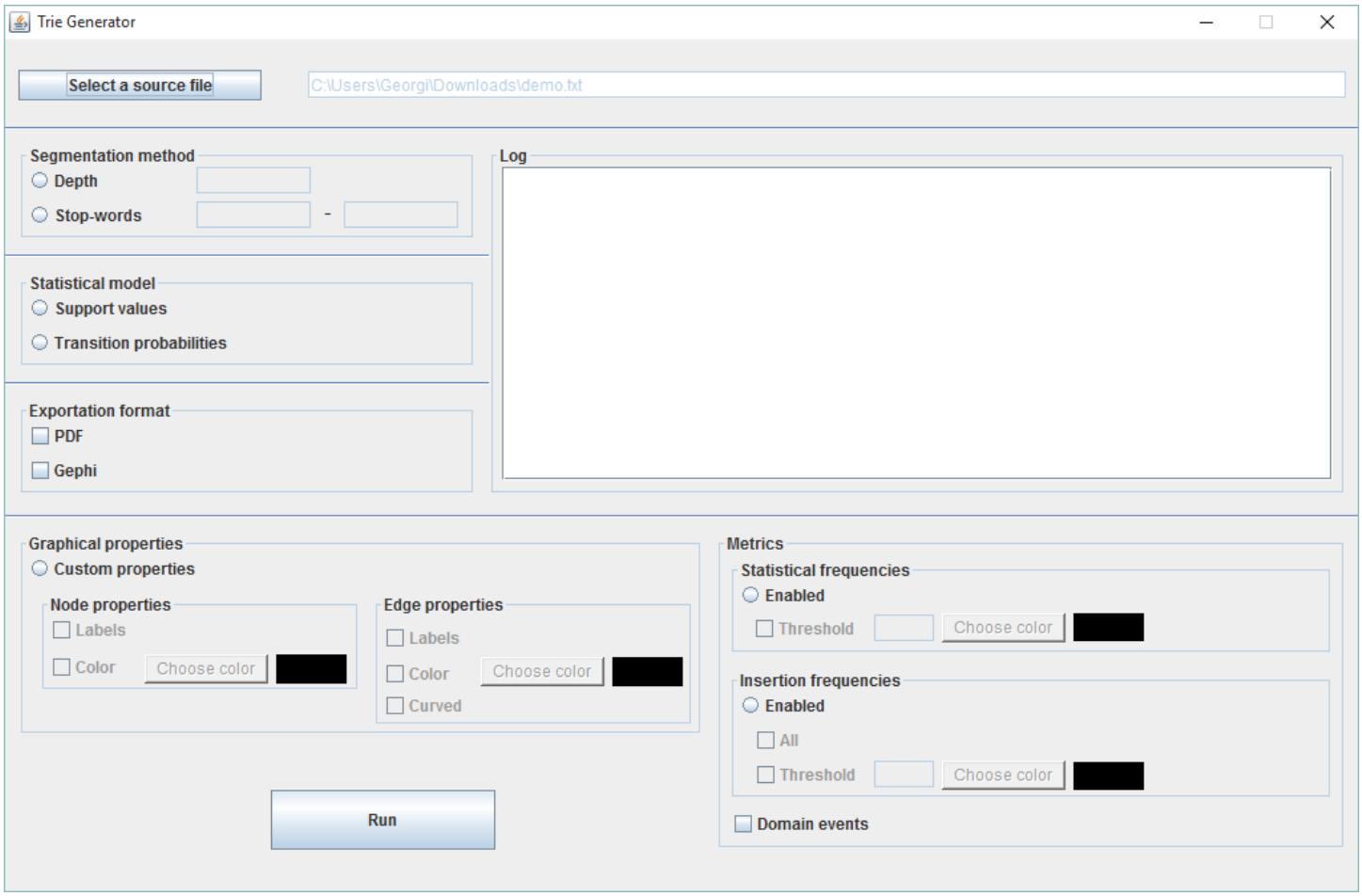

Ilustración 36: Manual de Usuario: ruta del fichero fuente.

1.3. Panel de Método de Segmentación

Este panel sirve para seleccionar el método de segmentación que se utilizará en la creación del trie. Existen dos opciones:

- **Profundidad**: este método de segmentación especifica la profundidad máxima que puede tener el trie. Como profundidad máxima se entiende la máxima longitud de una secuencia de eventos dentro del trie.
- **Palabras clave**: este método de segmentación controla la formación de las secuencias de eventos. La formación de las secuencias se realiza con las palabras clave indicadas, donde todas las secuencias de eventos empiezan y acaban con ellas.

A continuación se realiza un análisis más detallado de este panel:

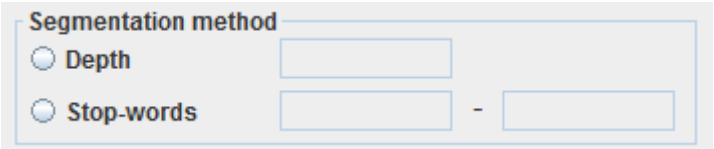

Ilustración 37: Manual de Usuario: método de segmentación.

Para seleccionar un método de segmentación es necesario pulsar sobre una de las dos opciones disponibles: Profundidad o Palabras clave (no es posible seleccionar ambas opciones). En función de la opción elegida, el campo de texto adjunto se habilitará (en el caso de las Palabras

clave, se habilitan los dos campos adjuntos. El primer campo especifica el inicio de las secuencias de eventos y el segundo campo especifica el fin de las secuencias de eventos).

1.4. Panel de Modelo Estadístico

En este panel se selecciona el modelo estadístico que se desea realizar en la formación del trie. Las dos opciones disponibles son:

- **Valores de soporte**: este modelo estadístico mide la relevancia de un evento dentro de todos los eventos de su mismo nivel o longitud.
- **Probabilidades de transición**: este modelo estadístico calcula la probabilidad de transitar de un evento a otro, en función del número de eventos que emanan del evento previo.

A continuación se presenta este panel con más detalle:

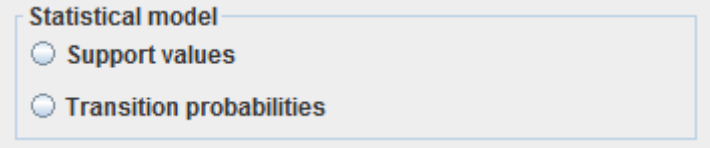

Ilustración 38: Manual de Usuario: modelo estadístico.

Para seleccionar un modelo es necesario pulsar sobre una de las dos opciones, ya que no es posible seleccionar ambas.

1.5. Panel de Exportación

Este panel se utiliza para seleccionar el formato de exportación de la representación gráfica del trie. Las dos opciones disponibles son PDF y Gephi.

A continuación se detalla el panel:

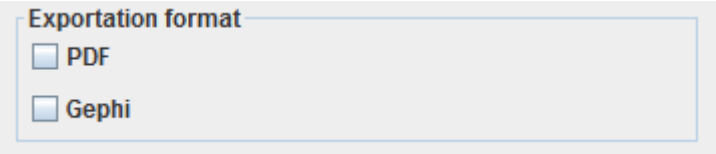

Ilustración 39: Manual de Usuario: formato de exportación.

Para seleccionar un formato de exportación se debe seleccionar cualquiera de las dos opciones, o incluso ambas.

1.6. Panel de Propiedades Gráficas

Este panel sirve para configurar todos los elementos del trie, tanto nodos, como enlaces. A continuación se detalla el panel:

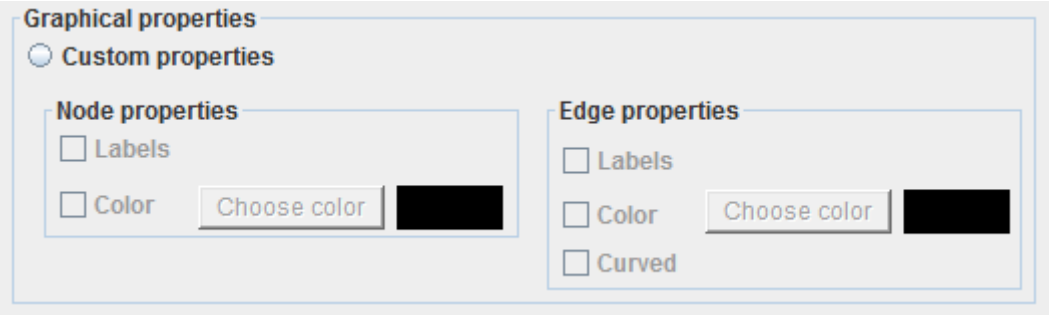

Ilustración 40: Manual de Usuario: propiedades gráficas.

Para activar o desactivar las propiedades gráficas se debe seleccionar la opción *Custom Properties*, situada en la esquina superior izquierda del panel. Cuando se activan las propiedades, todas las opciones, a excepción de los botones de selección de color se habilitan. Cuando se desactivan estas propiedades, todas las opciones se deshabilitan.

Este panel se compone de dos secciones:

- **Propiedades de los nodos**: esta sección permite configurar la aparición de las etiquetas de los nodos en la representación gráfica del trie. Además, se permite dar color a todos los nodos (a excepción del nodo raíz, que es siempre de color rojo). Para habilitar o deshabilitar las etiquetas de los nodos es necesario seleccionar la opción *Labels*. En cuanto al color de los nodos, es necesario en primer lugar seleccionar la opción *Color*. Una vez seleccionada, se habilita el botón de selección de color, *Choose color*. A continuación se ilustra la ventana que se abre cuando se pulsa el botón *Choose color*:

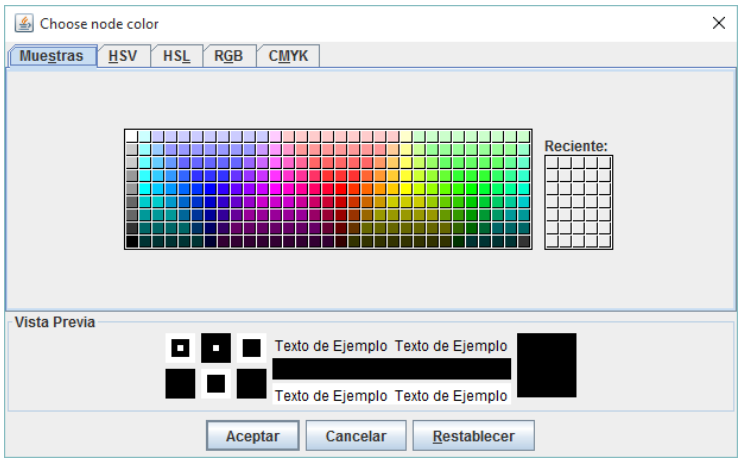

Ilustración 41: Manual de Usuario: selector de color.

Esta ventana se compone de cinco paletas de colores: Muestras, HSV, HSL, RGB y CMYK. Independientemente de la paleta de colores utilizada para seleccionar el color, en la sección de *Vista Previa*, situada en la parte inferior, se muestra la previsualización del color seleccionado. Una vez se ha elegido color, se debe pulsar *Aceptar* para confirmar el color. Una vez confirmado el color, este selector de colores se cerrará automáticamente.

- **Propiedades de los enlaces**: esta sección permite configurar la aparición de las etiquetas de los enlaces, el color de los enlaces y si los enlaces son o no curvos. Para habilitar o deshabilitar las etiquetas de los enlaces es necesario seleccionar la opción *Labels*. Las etiquetas de los enlaces (o pesos) se representan como la frecuencia estadística del nodo destino.
  En cuanto al color de los enlaces, es necesario seleccionar la opción *Color*. Cuando esta opción está seleccionada, se activa el botón *Choose color*. Para seleccionar el color de los enlaces se debe pulsar el botón *Choose color* y se desplegará la misma paleta de colores que se despliega en la selección de color de los nodos. Para confirmar el color de los enlaces es necesario pulsar sobre *Aceptar*, situado en la parte inferior derecha.
  Por último, para habilitar o deshabilitar la curvatura de los enlaces, es necesario seleccionar la opción *Curved*. Nótese que al habilitar esta opción se pierde la direccionalidad entre los nodos del trie.

1.7. Panel de Métricas

Este panel sirve para seleccionar las métricas que se desean aplicar sobre el trie. Estas métricas se agrupan en tres secciones:

- **Frecuencias estadísticas**: en esta sección se permite filtrar las frecuencias originadas en el modelo estadístico (ver 1.4. Panel de Modelo Estadístico). Esta métrica consiste en obtener todos los nodos cuyas frecuencias estadísticas superan o igualan el umbral indicado. El umbral debe ser un número entero o decimal comprendido entre 0 y 1, ambos incluidos.

  A continuación se detalla esta métrica:

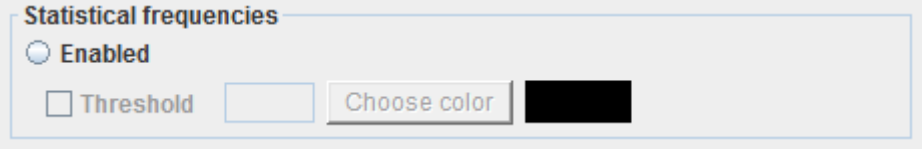

Ilustración 42: Manual de Usuario: frecuencias estadísticas.

  Para activar esta métrica es necesario seleccionar la opción *Enabled*. Una vez seleccionada esta opción se habilita la opción *Threshold*. En caso de activar esta opción, se habilita el campo de texto adjunto, junto con el botón *Choose color*. Al seleccionar el botón de selección de color se despliega el selector de color (véase 1.6. Panel de Propiedades Gráficas). Nótese que en la representación gráfica del trie, el color seleccionado en esta métrica afecta a los enlaces, cuyo peso es igual o superior al umbral indicado.
- **Frecuencias de inserción**: en esta sección se permite mostrar todas las frecuencias de inserción o filtrar aquellas que igualan o superan el umbral indicado. El umbral debe ser un número entero comprendido desde 0 hasta infinito. A continuación se detalla esta métrica:

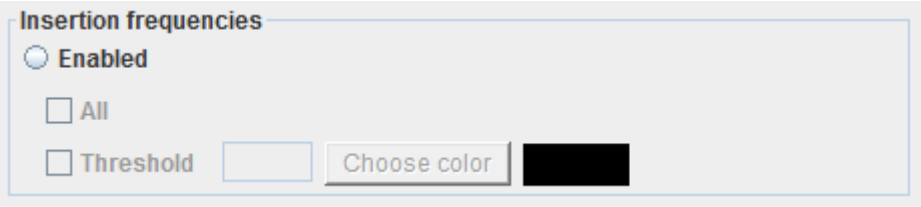

Ilustración 43: Manual de Usuario: frecuencias de inserción.

Para activar esta métrica es necesario seleccionar la opción *Enabled*. Una vez seleccionada esta opción se habilita la opción *All* y *Threshold*. La opción *All* se encarga de mostrar todos los nodos con sus frecuencias de inserción. La opción *Threshold* se encarga de filtrar aquello nodos cuyas frecuencias de inserción igualan o superan el umbral indicado. En caso de seleccionar la opción *Threshold* se habilita el campo de texto adjunto, junto con el botón *Choose color*. En caso de seleccionar el botón *Choose color* se despliega el selector de color (véase 1.6. Panel de Propiedades Gráficas). Nótese que en la representación gráfica del trie, el color seleccionado en esta métrica afecta a los nodos cuya frecuencia de inserción es igual o superior al umbral indicado.

- **Eventos del dominio**: esta métrica se encarga de generar los eventos individuales del dominio. A continuación se detalla esta métrica:

Ilustración 44: Manual de Usuario: eventos del dominio.

Para habilitar o deshabilitar esta métrica es necesario seleccionar la opción *Domain events*. Para entender esta métrica, véase **3.2.6.3. Eventos del dominio**.

1.8. Botón Run

Este botón sirve para ejecutar el sistema con todos los parámetros especificados previamente.

1.9. Log

Esta área de texto no editable muestra todos los procesos realizados en la creación, procesado y aplicación de métricas en la generación del trie resultante al ejecutar el programa. Los procesos que se muestran en el Log son:

- **Representación de ramas**: en este proceso se muestra el inicio de la ejecución y todas las ramas que va a tener el trie.

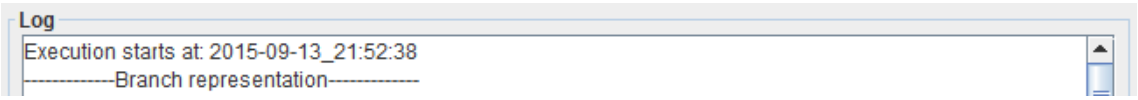

Ilustración 45: Manual de Usuario: Log - representación de ramas.

- **Creación de rutas**: en este proceso se crean todas las rutas de los nodos en el trie.

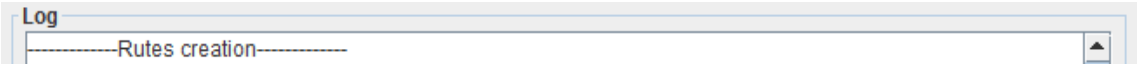

Ilustración 46: Manual de Usuario: Log - creación de rutas.

- **Creación de frecuencias**: en este proceso se calculan todas las frecuencias de inserción de los nodos en el trie.

Ilustración 47: Manual de Usuario: Log - creación de frecuencias.

- **Creación de nodos**: en este proceso se crean todos los nodos del trie, aplicándoles sus frecuencias de inserción e inicializando sus frecuencias estadísticas a 1.

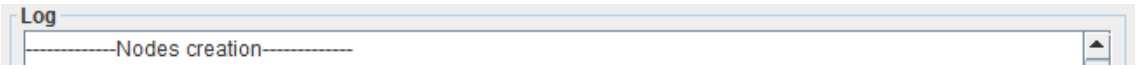

Ilustración 48: Manual de Usuario: Log - creación de nodos.

- **Tabla de valores de soporte**: en este proceso se crea una tabla intermedia para calcular los valores de soporte de los nodos, en caso de haber seleccionado este modelo estadístico. El contenido de esta tabla se estructura de la siguiente manera:

Nivel=total de nodos

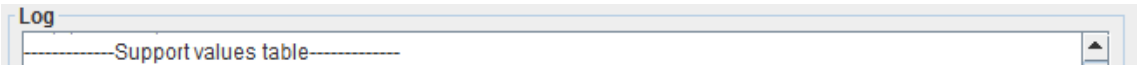

Ilustración 49: Manual de Usuario: Log - tabla de valores de soporte.

- **Nodos con valores de soporte**: en este proceso se establece la frecuencia estadística a los nodos como los valores de soporte, en caso de haber seleccionado este modelo estadístico.

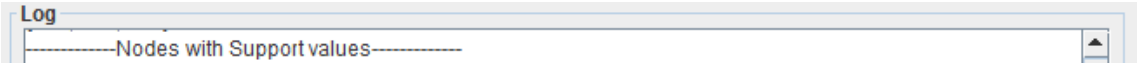

Ilustración 50: Manual de Usuario: Log - nodos con valores de soporte.

- **Tabla de probabilidades de transición**: en este proceso se crea una tabla intermedia para calcular las probabilidades de transición de los nodos, en caso de haber seleccionado este modelo estadístico. El contenido de esta tabla se estructura de la siguiente manera:

Nodo padre=número de nodos hijo

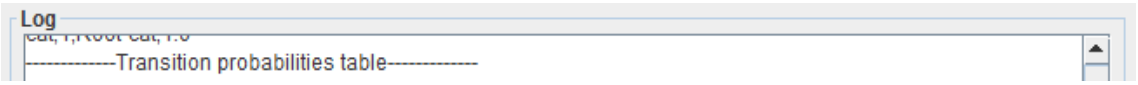

Ilustración 51: Manual de Usuario: Log - tabal de probabilidades de transición.

- **Nodos con probabilidades de transición**: en este proceso se establece la frecuencia estadística a los nodos como la probabilidad de transición entre ellos, en caso de haber seleccionado este modelo estadístico.

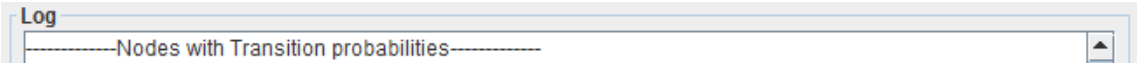

Ilustración 52: Manual de Usuario: Log - nodos con probabilidades de inserción.

- **Eventos del dominio**: en este proceso se listan todos los eventos del dominio.

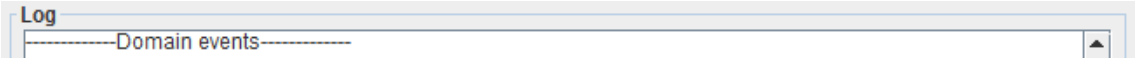

Ilustración 53: Manual de Usuario: Log - eventos del dominio.

- **Frecuencias estadísticas con umbral**: en este proceso se listan todos los nodos cuyas frecuencias estadísticas igualan o superan el umbral indicado.

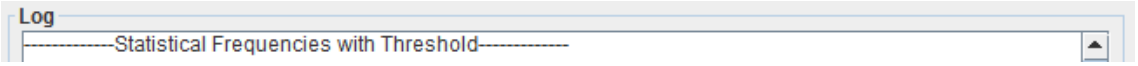

Ilustración 54: Manual de Usuario: Log - frecuencias estadísticas con umbral.

- **Todas las frecuencias de inserción**: en este proceso se listan todos los nodos con sus frecuencias de inserción.

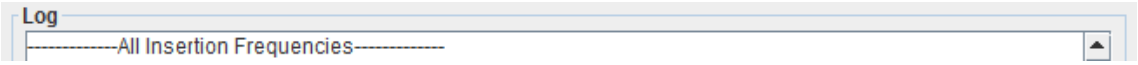

Ilustración 55: Manual de Usuario: Log - todas las frecuencias de inserción.

- **Frecuencias de inserción con umbral**: en este proceso se listan todos los nodos cuyas frecuencias de inserción iguala o supera el umbral indicado.

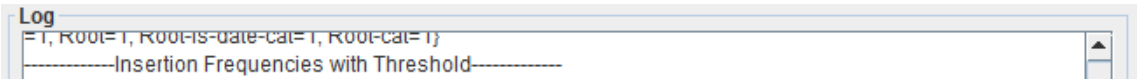

Ilustración 56: Manual de Usuario: Log - frecuencias de inserción con umbral.

- **Gephi**: este proceso muestra todo el procesado y pintado que realiza la librería Gephi para la representación gráfica del trie.

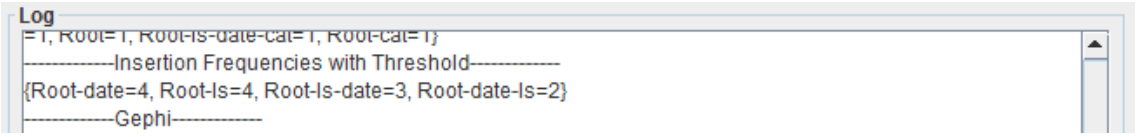

Ilustración 57: Manual de Usuario: Log - Gephi.

- **Número total de nodos y enlaces del trie**: este proceso muestra el número total de nodos y enlaces del trie generado.

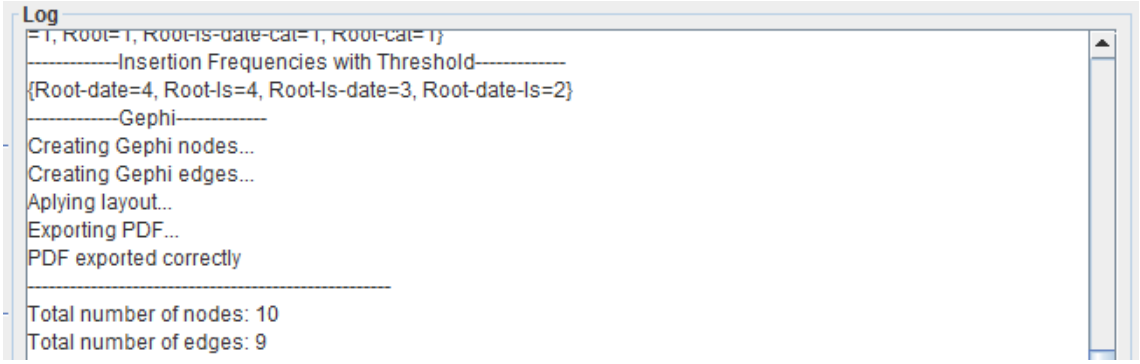

Ilustración 58: Manual de Usuario: Log - número total de nodos y enlaces del trie.

- **Tiempo total de ejecución**: este proceso muestra el tiempo de finalización del programa, así como el total de duración (en minutos y segundos).

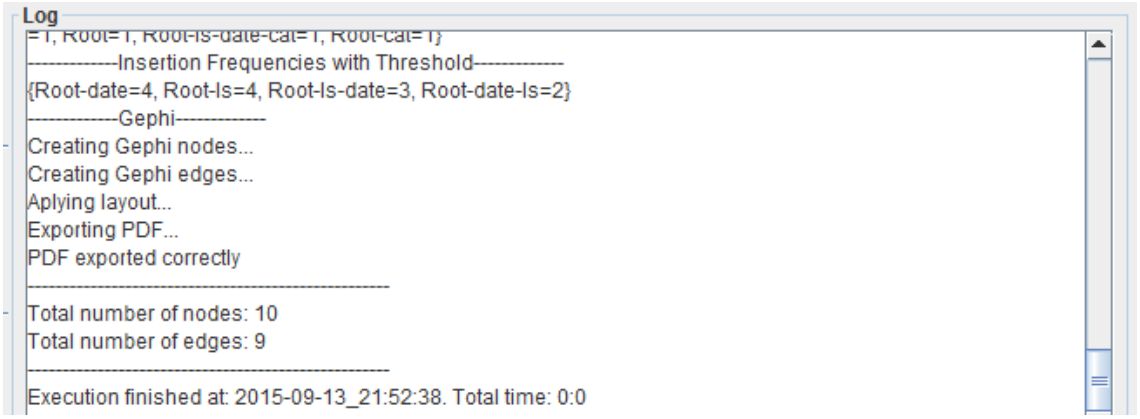

Ilustración 59: Manual de Usuario: Log - tiempo total de ejecución.

2. Ejecución

En esta sección del Manual de Usuario se detalla el procedimiento para ejecutar el programa y los ficheros que se generan tras su ejecución.

Para poder ejecutar el programa es necesario configurar los siguientes parámetros obligatorios:

- **Fichero fuente**: se debe seleccionar obligatoriamente un fichero fuente con eventos.
- **Método de segmentación**: se debe seleccionar obligatoriamente un método de segmentación de secuencias, junto con sus parámetros (nivel máximo de profundidad o palabras clave).
- **Modelo estadístico**: se debe seleccionar obligatoriamente un modelo estadístico para la creación del trie.
- **Formato de exportación**: se debe seleccionar obligatoriamente al menos un formato de exportación.

El resto de configuraciones son opcionales. Se debe tener en cuenta que si no se realizan métricas o se conilustraciónn propiedades gráficas, el trie resultante contará con todos los nodos y enlaces en color negro (a excepción del nodo *Root*, que tiene color rojo y su diámetro es el doble que le resto de nodos). El grosor de los enlaces varía en función de su peso: a mayor peso, mayor grosor.

Una vez conilustracióndos todos los parámetros obligatorios, es necesario pulsar el botón *Run* para ejecutar el programa.

Mientras se está ejecutando el programa, en el área del Log se muestran todas las trazas de los procesos intermedios a medida que se van completando.

Una vez acabado el proceso de análisis y formación del trie, en la carpeta en la que se sitúa el programa se generan varios ficheros, en función de las opciones y métricas elegidas:

- **Fichero Gephi**: este fichero contiene la representación gráfica del trie en formato Gephi. El fichero puede ser abierto con la aplicación de escritorio de Gephi. El nombre de este fichero sigue el siguiente formato:

  aaaa-mm-dd_hh-mm-ss.gexf

  , donde:
  - *aaaa* es el año
  - *mm* es el mes
  - *dd* es el día
  - *hh* es la hora (formato 24 horas)
  - *mm* son los minutos
  - *ss* son los segundos
- **Fichero PDF**: este fichero contiene la representación gráfica del trie en formato PDF. Se trata de una página de tamaño A4. El nombre de este fichero sigue el siguiente formato:

  aaaa-mm-dd_hh-mm-ss.pdf

  , donde:
  - *aaaa* es el año
  - *mm* es el mes
  - *dd* es el día
  - *hh* es la hora (formato 24 horas)
  - *mm* son los minutos
  - *ss* son los segundos
- **DomainEvents**: este fichero contiene todos los eventos del dominio, uno en cada fila del fichero. El nombre de este fichero sigue el siguiente formato:

  DomainEvents_aaaa-mm-dd_hh-mm-ss.txt

  , donde:
  - *aaaa* es el año
  - *mm* es el mes
  - *dd* es el día
  - *hh* es la hora (formato 24 horas)
  - *mm* son los minutos
  - *ss* son los segundos
- **InsertionFrequenciesAll**: este fichero contiene todos los nodos con sus frecuencias de inserción en el trie. El fichero se estructura en dos columnas separadas por una tabulación. En la primera columna se representan las frecuencias y en la segunda

columna se representa la ruta de cada nodo. El nombre de este fichero sigue el siguiente formato:

InsertionFrequenciesAll _aaaa-mm-dd_hh-mm-ss.txt

, donde:

- *aaaa* es el año
- *mm* es el mes
- *dd* es el día
- *hh* es la hora (formato 24 horas)
- *mm* son los minutos
- *ss* son los segundos

- **InsertionFrequenciesWithThreshold**: este fichero contiene todos los nodos que tienen una frecuencia de inserción igual o superior al umbral indicado en la ejecución que lo ha generado. El fichero se estructura en dos columnas separadas por una tabulación. En la primera columna se representan las frecuencias y en la segunda columna se representa la ruta de cada nodo. El nombre de este fichero sigue el siguiente formato:

  InsertionFrequenciesWithThreshold _aaaa-mm-dd_hh-mm-ss.txt

  , donde:

  - *aaaa* es el año
  - *mm* es el mes
  - *dd* es el día
  - *hh* es la hora (formato 24 horas)
  - *mm* son los minutos
  - *ss* son los segundos
- **StatisticalFrequenciesWithThreshold**: este fichero contiene todos los nodos que tienen una frecuencia estadística igual o superior al umbral indicado en la ejecución que lo generado. El fichero se estructura en dos columnas separadas por una tabulación. En la primera columna se representan las frecuencias y en la segunda columna se representa la ruta de cada nodo. El nombre de este fichero sigue el siguiente formato:

  StatisticalFrequenciesWithThreshold _aaaa-mm-dd_hh-mm-ss.txt

  , donde:

  - *aaaa* es el año
  - *mm* es el mes
  - *dd* es el día
  - *hh* es la hora (formato 24 horas)
  - *mm* son los minutos
  - *ss* son los segundos
- **SupportValues**: este fichero contiene todos los nodos con sus valores de soporte, ordenados en forma descendente en función de sus valores de soporte. El fichero se estructura en dos columnas separadas por una tabulación. En la primera columna se representan los valores de soporte y en la segunda columna se representa la ruta de cada nodo. El nombre de este fichero sigue el siguiente formato:

  SupportValues _aaaa-mm-dd_hh-mm-ss.txt

  , donde:

  - *aaaa* es el año
  - *mm* es el mes
  - *dd* es el día
  - *hh* es la hora (formato 24 horas)
  - *mm* son los minutos

- *ss* son los segundos

- **TrieData**: este fichero contiene el número total de nodos y en enlaces del trie, así como todos sus nodos, de los que se representan todos sus atributos separados por comas. Estos atributos son:
  - Contenido
  - Frecuencia de inserción
  - Ruta
  - Frecuencia estadística

  El nombre de este fichero sigue el siguiente formato:

  TrieData _aaaa-mm-dd_hh-mm-ss.txt

  , donde:
  - *aaaa* es el año
  - *mm* es el mes
  - *dd* es el día
  - *hh* es la hora (formato 24 horas)
  - *mm* son los minutos
  - *ss* son los segundos

3. Tratamiento de errores

En esta sección del Manual de Usuario se detallan todo el tratamiento de errores que posee el sistema. A continuación se detallan todos los casos de error que soporta el sistema:

- No se ha especificado uno o más de los parámetros obligatorios necesarios para la ejecución del sistema. En la siguiente ilustración se muestran todos los errores cuando no se ha especificado ningún parámetro obligatorio:

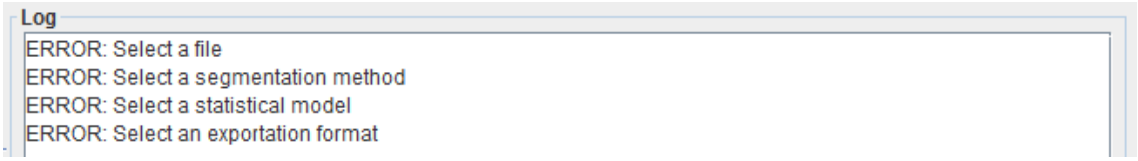

Ilustración 60: Manual de Usuario: Tratamiento de errores - parámetros obligatorios.

En caso de ocurrir este error, el programa se detiene y no se genera ningún fichero.

- Se ha introducido un fichero vacío:

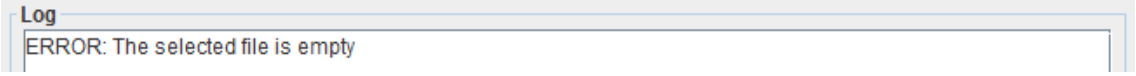

Ilustración 61: Manual de Usuario: Tratamiento de errores - fichero vacío.

En caso de ocurrir este error, el programa se detiene y no se genera ningún fichero.

- Se ha introducido un fichero que contiene el carácter " – ":

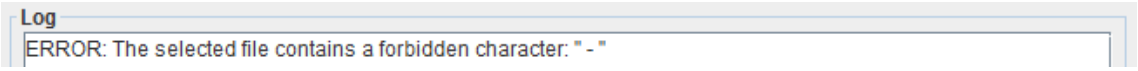

Ilustración 62: : Manual de Usuario: Tratamiento de errores - carácter inválido.

Este error se presenta debido a que el carácter " – " es utilizado internamente para el procesado de las secuencias de eventos. Este carácter debe ser sustituido por otro para poder ejecutar el programa con ese fichero. En caso de ocurrir este error, el programa se detiene y no se genera ningún fichero.

- No se ha especificado un valor para un campo:

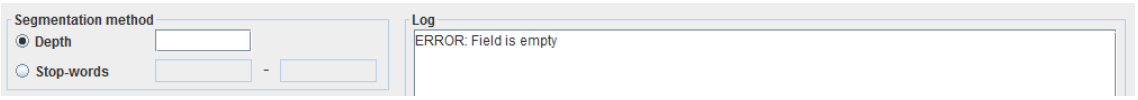

Ilustración 63: Manual de Usuario: Tratamiento de errores - campo vacío.

Este error se presenta en todos los campos en los que no se ha rellenado nada: nivel de profundidad, alguna de las dos palabras clave (o las dos), valores de threshold, tanto de las frecuencias estadísticas, como de las frecuencias de inserción. En caso de ocurrir este error, el programa se detiene y no se genera ningún fichero.

- Se ha especificado un valor incorrecto para la profundidad en el Método de segmentación:

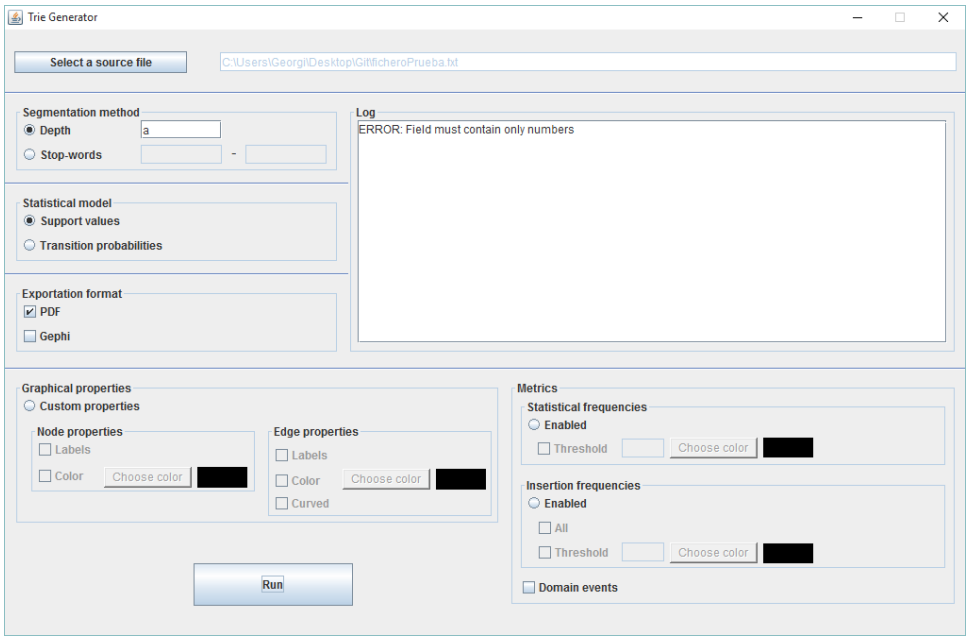

Ilustración 64: Manual de Usuario: Tratamiento de errores - profundidad incorrecta.

En caso de ocurrir este error, el programa se detiene y no se genera ningún fichero.

- Se ha introducido una profundidad tan larga que supera el total de longitud de la secuencia de eventos del fichero fuente:

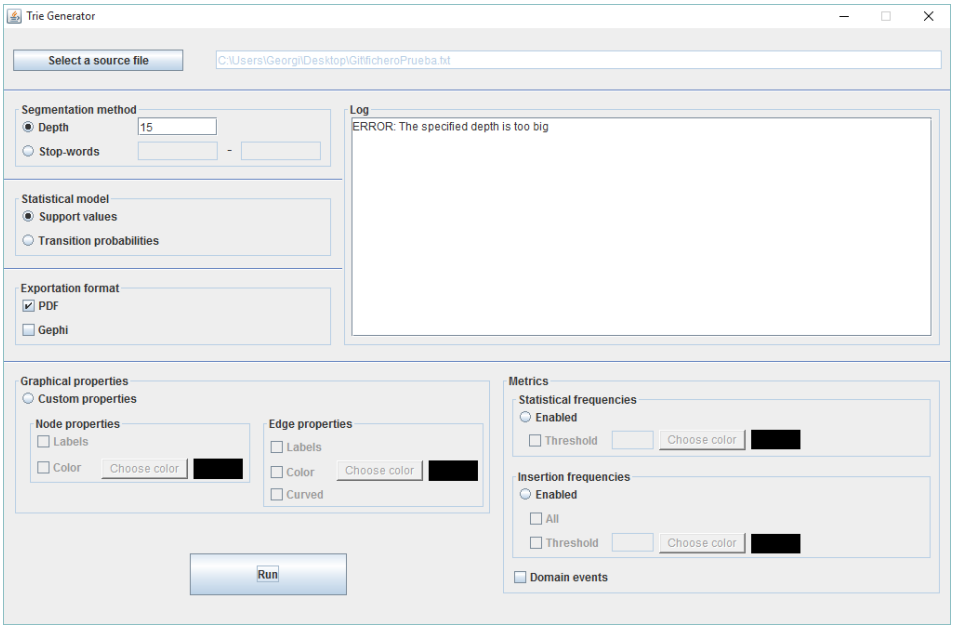

Ilustración 65: Manual de Usuario: Tratamiento de errores - profundidad demasiado grande.

En caso de ocurrir este error, el programa se detiene y no se genera ningún fichero.

- Se ha introducido una palabra clave inexistente respecto a los eventos del fichero fuente:

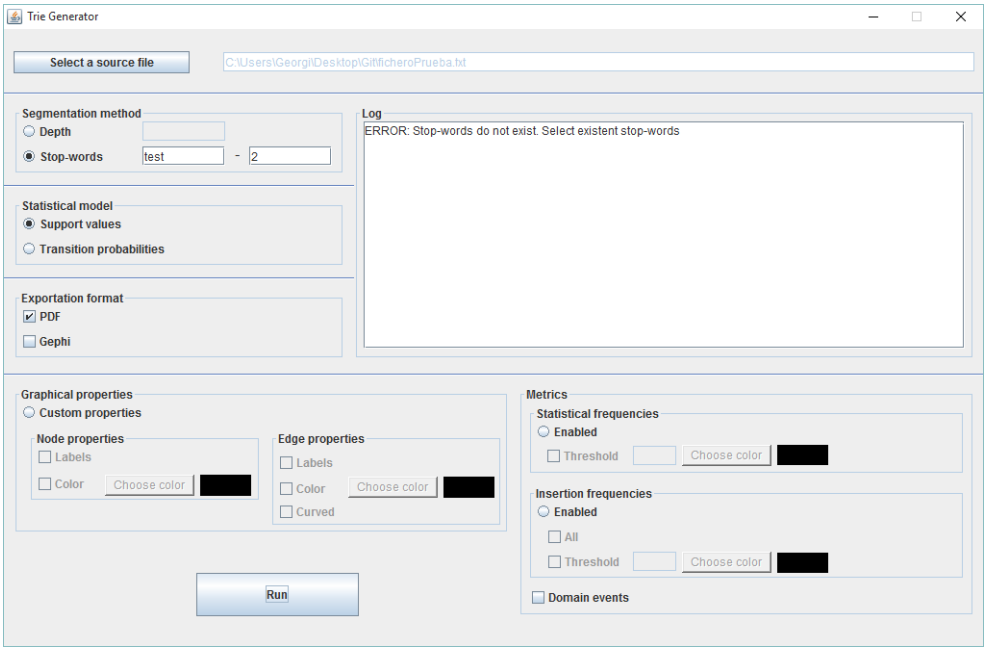

Ilustración 66: Manual de Usuario: Tratamiento de errores - palabra clave inexistente.

En caso de ocurrir este error, el programa se detiene y no se genera ningún fichero.

- Se ha introducido un valor de umbral incorrecto en la métrica *Frecuencias estadísticas*:

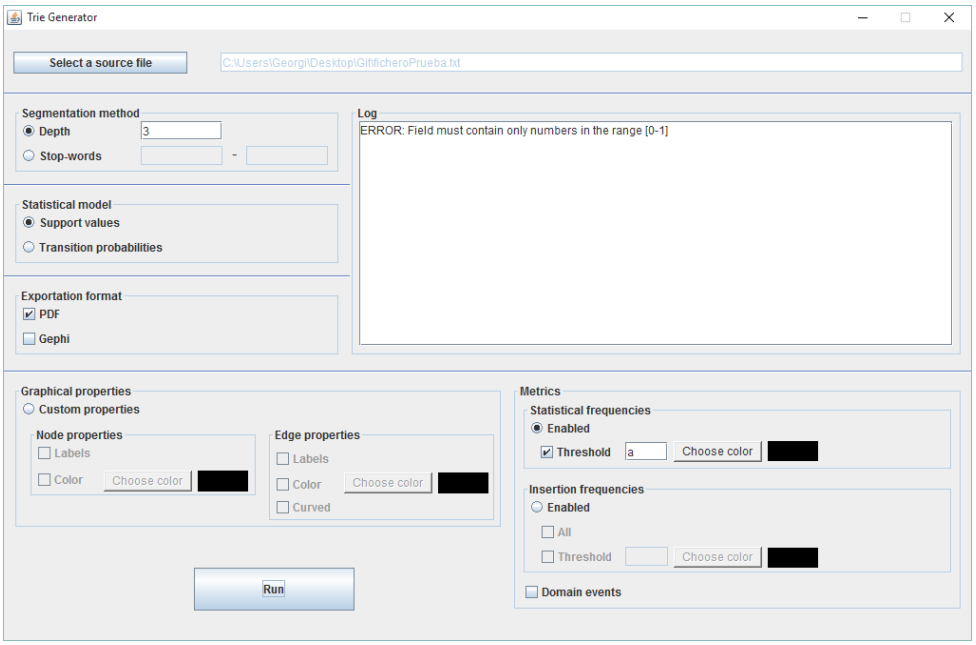

Ilustración 67: Manual de Usuario: Tratamiento de errores - umbral incorrecto en Frecuencias estadísticas.

En caso de ocurrir este error, el programa se detiene y no se genera ningún fichero.

- Se ha introducido un valor de umbral incorrecto en la métrica *Frecuencias de inserción*:

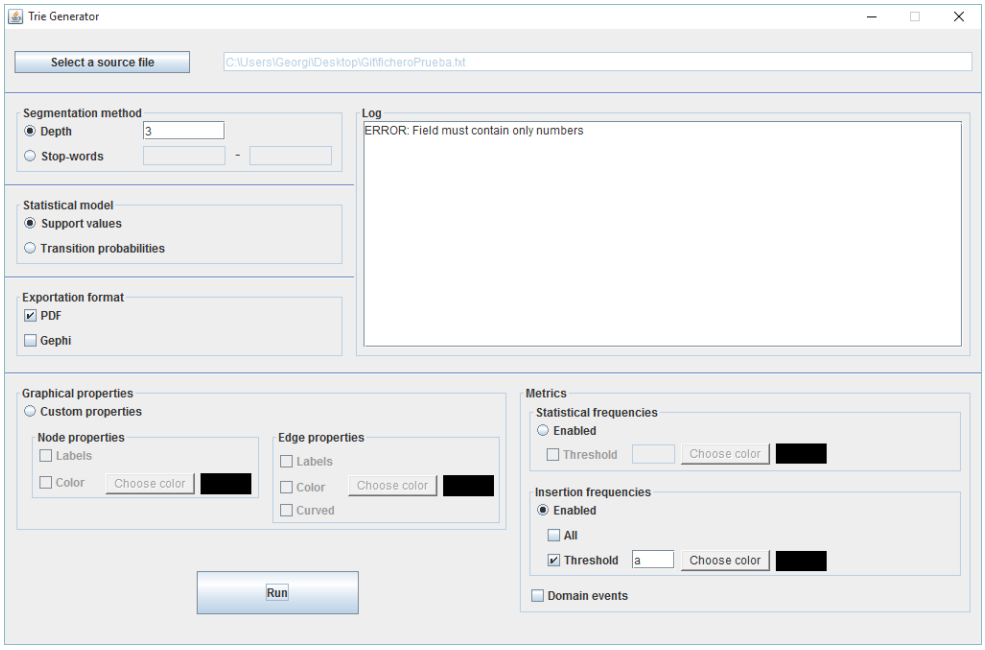

Ilustración 68: Manual de Usuario: Tratamiento de errores - umbral incorrecto en Frecuencias de inserción.

En caso de ocurrir este error, el programa se detiene y no se genera ningún fichero.

- Se ha introducido un valor de umbral demasiado elevado en la métrica *Frecuencias estadísticas* o *Frecuencias de inserción*. Este error se debe a que no existe ningún nodo del trie que tenga una frecuencia tan elevada (estadística o de inserción). Nótese que la ejecución del programa no se ve afectada por este error. La ejecución se completa, y cuando se procesa esta métrica, aparece un mensaje en el Log notificando el error:

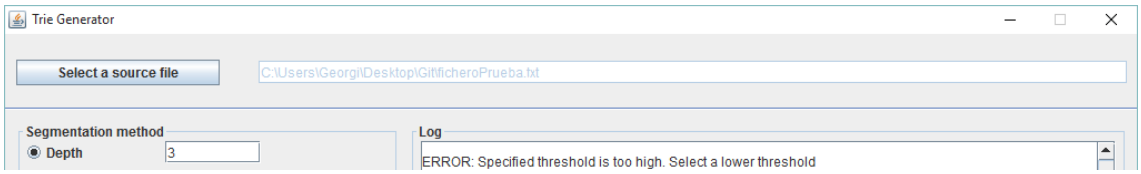

Ilustración 69: Manual de Usuario: Tratamiento de errores - umbral demasiado elevado.

El fichero asociado a esta métrica no se genera en caso de error.

Anexo III: Introduction

1. Motivation

Agent modeling is defined as *the ability of acquiring, inferring and storing the knowledge (behavior, believes, goals, actions, plans…) from other agents* [1]. The goal of the agent modeling is to infer what an agent is doing or what it intends to do, considering the actions it performed or the events that happened in its environment. It is considered as agent, a human being, a software system or a hardware system.

When the agent is a human being, there is a wide research field around the human behavior study. One are of interest in the study of human behavior is marketing. Knowing customers and their preferences about a certain product is a subject of great interest in this area, which seeks to increase the demand of products aimed to certain type of consumers. An example of the application of Business Intelligence can be found in commerce companies like Amazon, which uses this kind of techniques to analyze their clients and make predictions, based on their behaviors, of products that could potentially be bought by these clients. Thus, it is been created a recommendation system that generates important revenue in the business model that follows Amazon [2].

In the past few years, thanks to the advances in robotics and artificial intelligence, there are more and more intelligent systems that can communicate with a person. These systems communicate with people using graphical interfaces, sounds and even using their physical extensions of their own body, in the case of robots. Building friendly and functional systems which can interact with any user, regardless of their education and specialization is a topic under investigation and gradually prototypes that try to emulate full human behavior are arising. An example of this are the robot assistants. Examples of these kind of agents are NAO [3] and ASIMO [4]. In the case of ASIMO, it is able to see, to speak, to pick objects, to run and climb stairs. The communication with it can be done through speech, making speech understanding key part of the communication with the robot.

The construction of systems based on the behavior of an agent can be found in many other areas, such as the videogame industry. In this area, the artificial intelligence of the game must adapt to the player's behavior, to offer a challenging experience to the intelligence and ability that the player manifests. An example of this can be found in [5], where World of Warcraft players' behavior is modeled.

With today's technological advances and the worldwide Internet expansion, the security and protection of computer systems has become a priority. Cyber-attacks that aim to invade privacy or damage a system are a priority for ordinary users, companies and even governments. The study of malicious agents in these kind of attacks contributes to the creation of defenses, both hardware and software, against cyber-attacks. There is extensive literature about the behavior of a malicious agents inside a system, such as in [6], where behavior modeling is used to identify the components of a malicious software.

The motivation of this project is to contribute to agent modeling an automatic system that facilitates the analysis of sequential behavior.

2. Goals

In this section are described the goals, both main and specific, that the project aims to achieve.

2.1. Main goal

The main goal of this project is the development of a software tool, domain-independent for the analysis of action sequences, using tries like representation method.

2.2. Specific goals

To achieve the main goal of this project, the following specific goals must be achieved first:

- To provide the system with a user interface, where the analyst can set all the configuration and representation parameters of the trie.
- To design the system to be capable of operating independently of the studied domain, and to process any type of action or event sequences.
- To get various statistical metrics focused on the sequence analysis and pattern extraction. These metrics should be:
    - Support values.
    - Domain events.
    - Action filtering with insertion frequencies in the trie with a minimum threshold.
    - Action filtering with statistical frequencies in the trie with a minimum threshold.
- To represent graphically the resulting trie imposed by the analyst who uses the tool.

3. Structure of the document

In this section it's described the structure that presents the document:

Chapter 1: the motivation to make this project like *Trabajo Fin de Grado* is presented. In addition, the main goal and specific goals needed to achieve are described in this section.

Chapter 2: this section covers the state of art in which this research is framed, defining concepts and showing parallel researches to the goals of this project.

Chapter 3: in this section, a detailed description of the implemented system is done, reviewing all execution flows and constraints under which the system operates. In addition, the used technologies and the reasons of their usage in this project are explained.

Chapter 4: represents the performed experimentation with the implemented system. In the section, the data domain and data set used to verify the performance and utility of the systems are described.

Chapter 5: this section discusses the development of the project, exposing the followed planning and the necessary budget to take the system to a commercial level.

Chapter 6: the conclusions of the experimentation and the project are exposed. In addition, future work, based on the research performed in this project, is described.

Annexes: the last part of the document is composed by a collection of annexes. In the first annex, the results of the experimentation, omitted in chapter 4, are shown. The second annex contains the User Manual, which describes the structure, the execution and the troubleshooting processes which the tool possesses. Finally, the annexes three, four and five contain the English translations of the introduction, the experimentation results and the conclusions and future work.

Anexo IV: Experimentation

1. Results

This section contains the obtained results after the experimentation stage. To evaluate the results, the 20 most relevant commands from each user at the first three depths have been selected. This set of commands represents the model of each user.

In order to show these commands as vividly as possible, the results of each user are displayed in a table and a bar graph associated with this table. The graphical representation of the obtained tries has been omitted, because of the large amount of nodes and links that they have, making them undistinguishable in A4 sheet size. However, the Gephi desktop application used in this project allows to explore accurately the obtained tries, allowing researchers to navigate through each node and each link.

User 1

Depth 3	
Support value	Node
0,278	Root-netscape
0,252	Root-netscape-netscape
0,23	Root-netscape-netscape-netscape
0,078	Root-sh
0,043	Root-csh
0,042	Root-egrep
0,04	Root-sendmail
0,033	Root-egrep-egrep
0,03	Root-cat
0,029	Root-launchef
0,025	Root-egrep-egrep-egrep
0,024	Root-expr
0,023	Root-generic
0,023	Root-ls
0,016	Root-MediaMai
0,016	Root-sendmail-sendmail
0,015	Root-launchef-sh
0,015	Root-rm
0,015	Root-sh-MediaMai
0,015	Root-java

Tabla 101: Support values for User 1 with Depth 3.

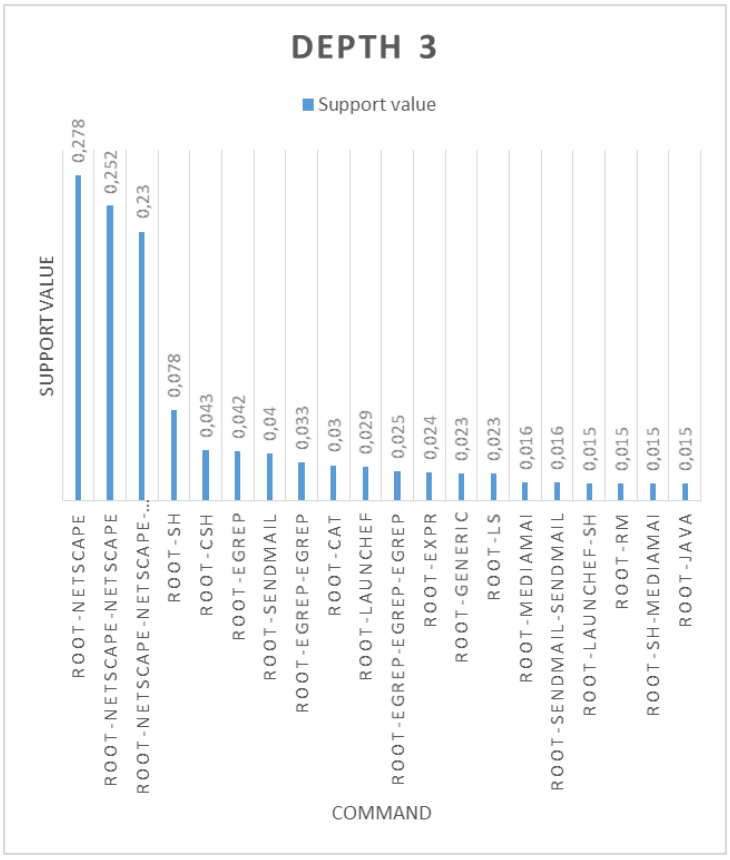

Ilustración 70: Graph with support values for User 1 with Depth 3.

Depth 5	
Support value	Node
0,278	Root-netscape
0,252	Root-netscape-netscape
0,23	Root-netscape-netscape-netscape
0,212	Root-netscape-netscape-netscape-netscape
0,195	Root-netscape-netscape-netscape-netscape-netscape
0,078	Root-sh
0,043	Root-csh
0,042	Root-egrep
0,04	Root-sendmail
0,033	Root-egrep-egrep
0,03	Root-cat
0,029	Root-launchef
0,025	Root-egrep-egrep-egrep
0,024	Root-expr
0,023	Root-generic
0,023	Root-ls
0,016	Root-MediaMai
0,016	Root-sendmail-sendmail
0,016	Root-egrep-egrep-egrep-egrep
0,015	Root-launchef-sh

Tabla 102: Support values for User 1 with Depth 5.

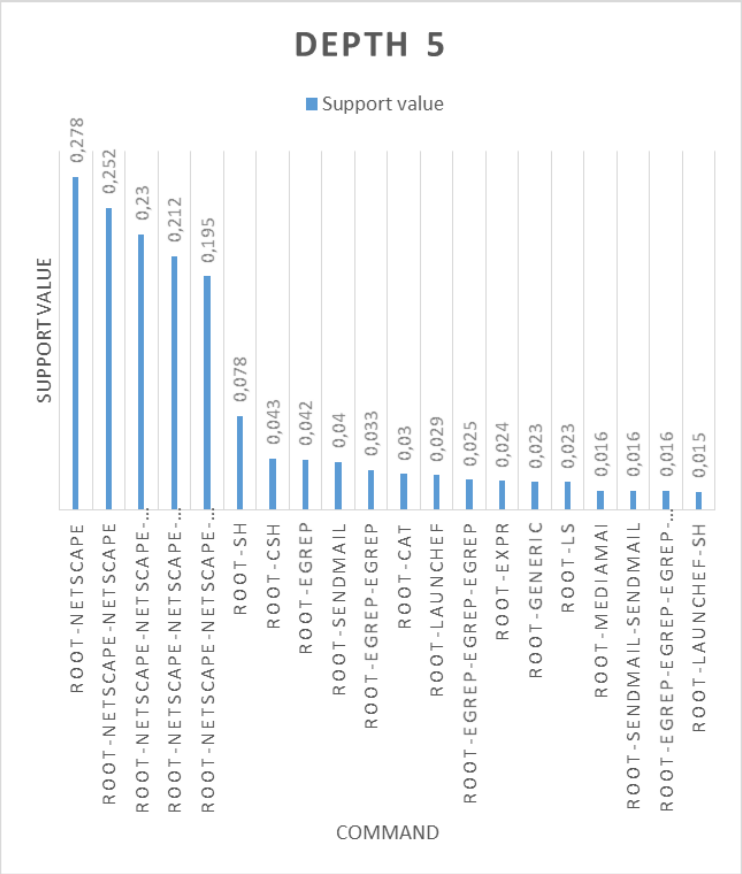

Ilustración 71: Graph with support values for User 1 with Depth 5.

Depth 7	
Support value	Node
0,278	Root-netscape
0,252	Root-netscape-netscape
0,23	Root-netscape-netscape-netscape
0,212	Root-netscape-netscape-netscape-netscape
0,196	Root-netscape-netscape-netscape-netscape-netscape
0,18	Root-netscape-netscape-netscape-netscape-netscape-netscape
0,165	Root-netscape-netscape-netscape-netscape-netscape-netscape-netscape
0,078	Root-sh
0,043	Root-csh
0,042	Root-egrep
0,04	Root-sendmail
0,033	Root-egrep-egrep
0,03	Root-cat
0,029	Root-launchef
0,024	Root-expr
0,024	Root-egrep-egrep-egrep
0,023	Root-generic
0,023	Root-ls
0,016	Root-MediaMai
0,016	Root-sendmail-sendmail

Tabla 103: Support values for User 1 with Depth 7.

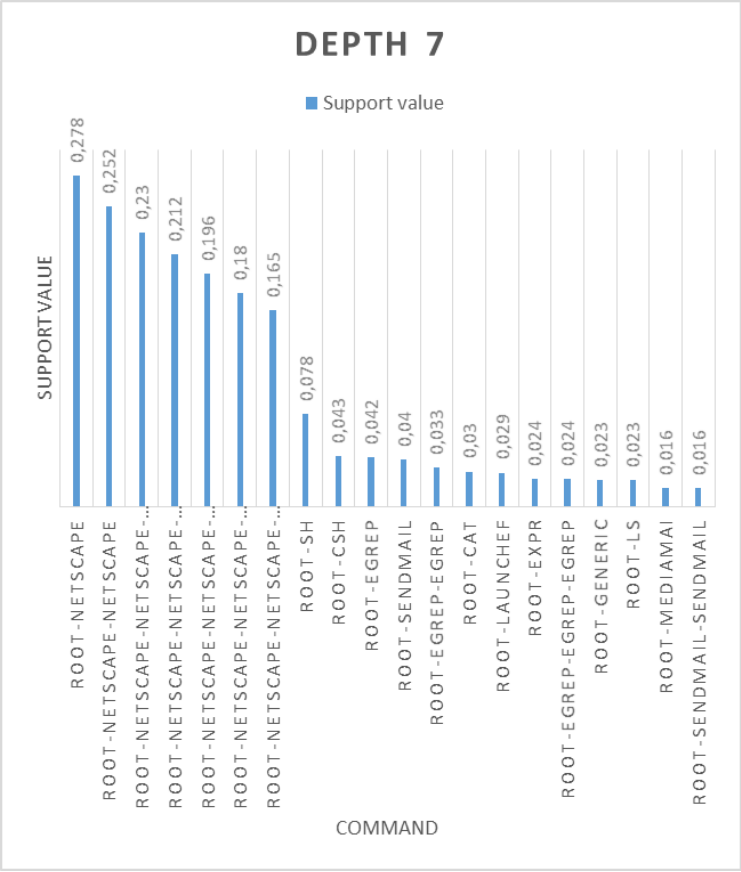

Ilustración 72: Graph with support values for User 1 with Depth 7.

User 2

Depth 3	
Support value	Node
0,138	Root-gcc
0,091	Root-awk
0,081	Root-less
0,070	Root-nawk
0,069	Root-uname
0,069	Root-uname-nawk
0,069	Root-gcc-gcc
0,057	Root-awk-less
0,053	Root-uname-nawk-cpp
0,053	Root-nawk-cpp
0,053	Root-cpp
0,052	Root-nawk-cpp-cc1
0,052	Root-cpp-cc1
0,052	Root-cc1
0,044	Root-cpp-cc1-as
0,044	Root-cc1-as
0,044	Root-as
0,043	Root-make
0,042	Root-cat
0,036	Root-less-awk

Tabla 104: Support values for User 2 with Depth 3.

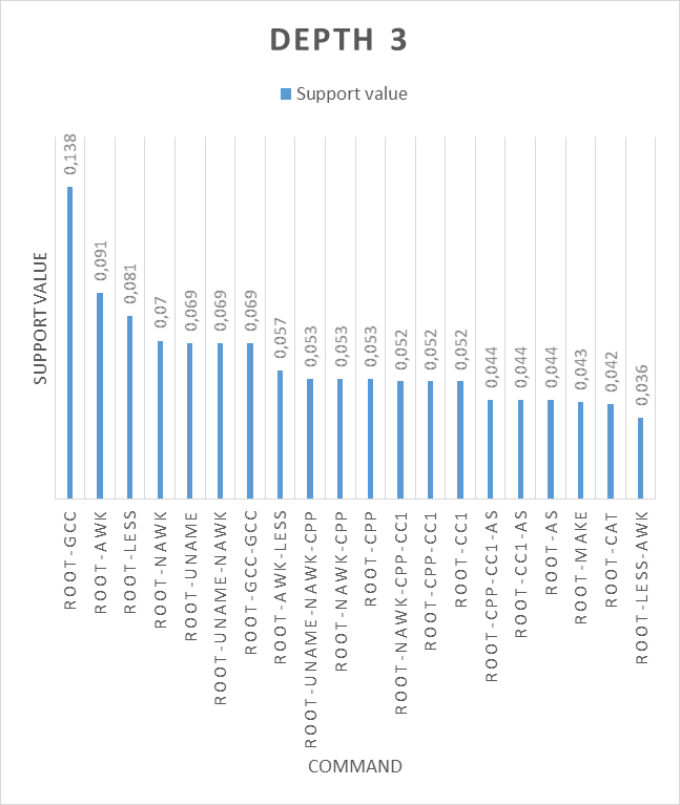

Ilustración 73: Graph with support values for User 2 with Depth 3.

Depth 5	
Support value	Node
0,138	Root-gcc
0,091	Root-awk
0,081	Root-less
0,070	Root-nawk
0,069	Root-uname
0,069	Root-uname-nawk
0,069	Root-gcc-gcc
0,057	Root-awk-less
0,053	Root-uname-nawk-cpp
0,053	Root-nawk-cpp
0,053	Root-cpp
0,052	Root-uname-nawk-cpp-cc1
0,052	Root-nawk-cpp-cc1
0,052	Root-cpp-cc1
0,052	Root-cc1
0,044	Root-uname-nawk-cpp-cc1-as
0,044	Root-nawk-cpp-cc1-as
0,044	Root-cpp-cc1-as
0,044	Root-cc1-as
0,044	Root-as

Tabla 105: Support values for User 2 with Depth 5.

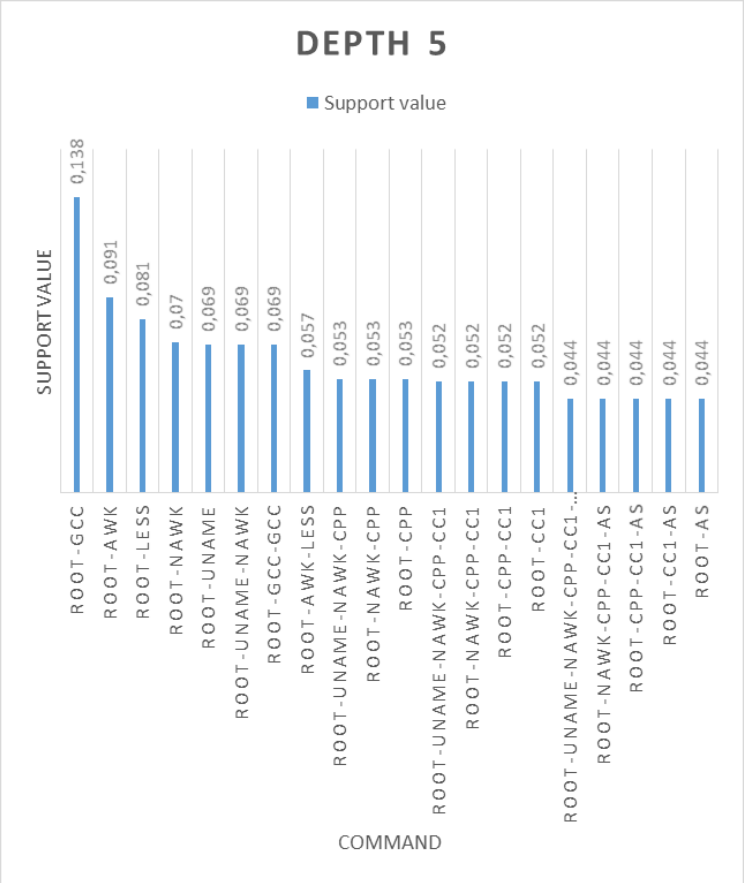

Ilustración 74: Graph with support values for User 2 with Depth 5.

Depth 7	
Support value	Node
0,138	Root-gcc
0,091	Root-awk
0,081	Root-less
0,070	Root-nawk
0,069	Root-uname
0,069	Root-uname-nawk
0,069	Root-gcc-gcc
0,057	Root-awk-less
0,053	Root-uname-nawk-cpp
0,053	Root-nawk-cpp
0,053	Root-cpp
0,052	Root-uname-nawk-cpp-cc1
0,052	Root-nawk-cpp-cc1
0,052	Root-cpp-cc1
0,052	Root-cc1
0,044	Root-uname-nawk-cpp-cc1-as
0,044	Root-nawk-cpp-cc1-as
0,044	Root-cpp-cc1-as
0,044	Root-cc1-as
0,044	Root-as

Tabla 106: Support values for User 2 with Depth 7.

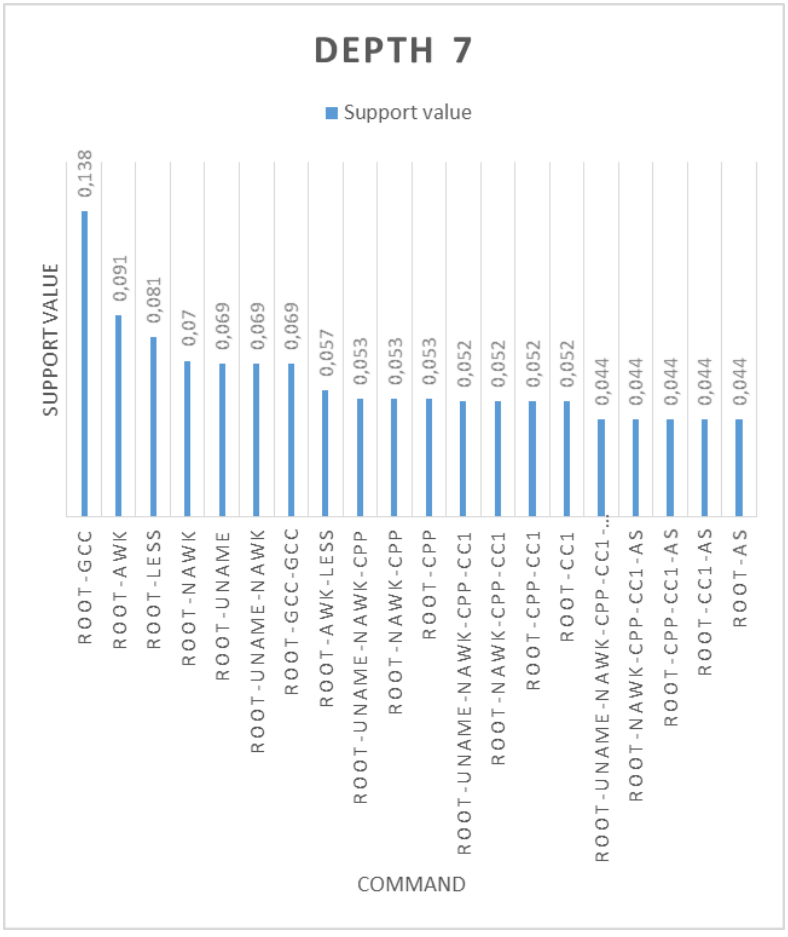

Ilustración 75: Graph with support values for User 2 with Depth 7.

User 3

Depth 3	
Support value	Node
0,174	Root-egrep
0,154	Root-expr
0,130	Root-egrep-egrep
0,092	Root-expr-expr
0,087	Root-egrep-egrep-egrep
0,087	Root-java
0,061	Root-expr-expr-dirname
0,061	Root-expr-dirname
0,061	Root-dirname
0,044	Root-basename
0,043	Root-.java_wr
0,043	Root-.java_wr-expr
0,043	Root-.java_wr-expr-expr
0,043	Root-expr-dirname-basename
0,043	Root-dirname-basename
0,043	Root-dirname-basename-egrep
0,043	Root-basename-egrep
0,043	Root-basename-egrep-egrep
0,042	Root-java-java
0,036	Root-make

Tabla 107: Support values for User 3 with Depth 3.

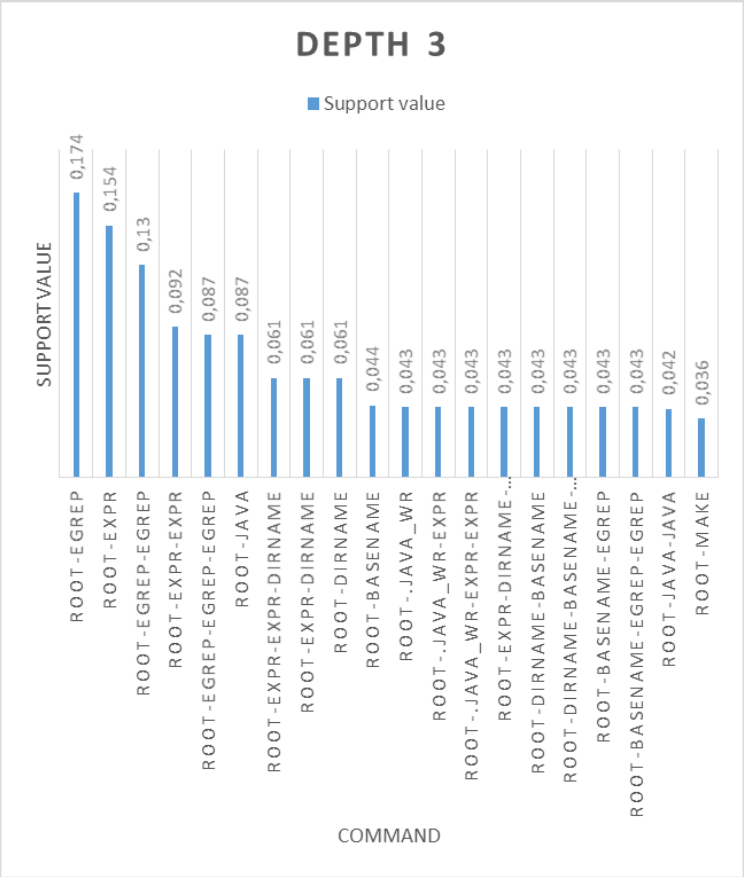

Ilustración 76: Graph with support values for User 3 with Depth 3.

Depth 5	
Support value	Node
0,174	Root-egrep
0,154	Root-expr
0,130	Root-egrep-egrep
0,092	Root-expr-expr
0,087	Root-egrep-egrep-egrep
0,087	Root-java
0,061	Root-expr-expr-dirname
0,061	Root-expr-dirname
0,061	Root-dirname
0,044	Root-basename
0,043	Root-.java_wr
0,043	Root-.java_wr-expr
0,043	Root-.java_wr-expr-expr
0,043	Root-.java_wr-expr-expr-dirname
0,043	Root-.java_wr-expr-expr-dirname-basename
0,043	Root-expr-expr-dirname-basename
0,043	Root-expr-dirname-basename
0,043	Root-dirname-basename
0,043	Root-expr-expr-dirname-basename-egrep
0,043	Root-expr-dirname-basename-egrep

Tabla 108: Support values for User 3 with Depth 5.

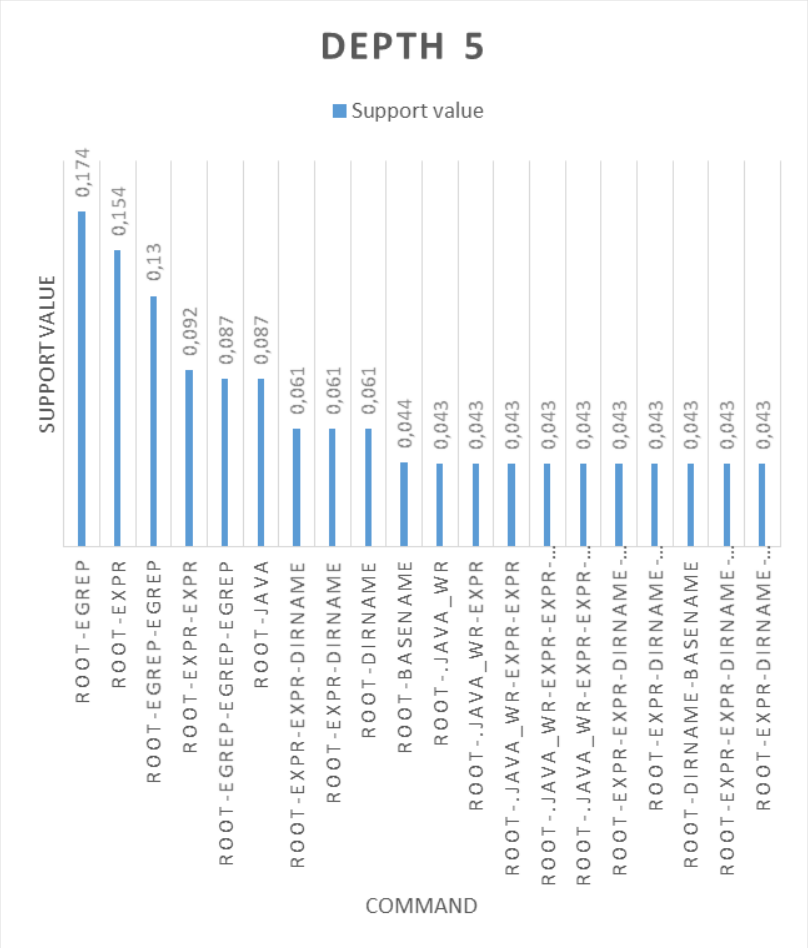

Ilustración 77: Graph with support values for User 3 with Depth 5.

Depth 7	
Support value	Node
0,174	Root-egrep
0,154	Root-expr
0,130	Root-egrep-egrep
0,092	Root-expr-expr
0,087	Root-egrep-egrep-egrep
0,087	Root-java
0,061	Root-expr-expr-dirname
0,061	Root-expr-dirname
0,061	Root-dirname
0,044	Root-basename
0,043	Root-.java_wr
0,043	Root-.java_wr-expr
0,043	Root-.java_wr-expr-expr
0,043	Root-.java_wr-expr-expr-dirname
0,043	Root-.java_wr-expr-expr-dirname-basename
0,043	Root-expr-expr-dirname-basename
0,043	Root-expr-dirname-basename
0,043	Root-dirname-basename
0,043	Root-.java_wr-expr-expr-dirname-basename-egrep
0,043	Root-expr-expr-dirname-basename-egrep

Tabla 109: Support values for User 3 with Depth 7.

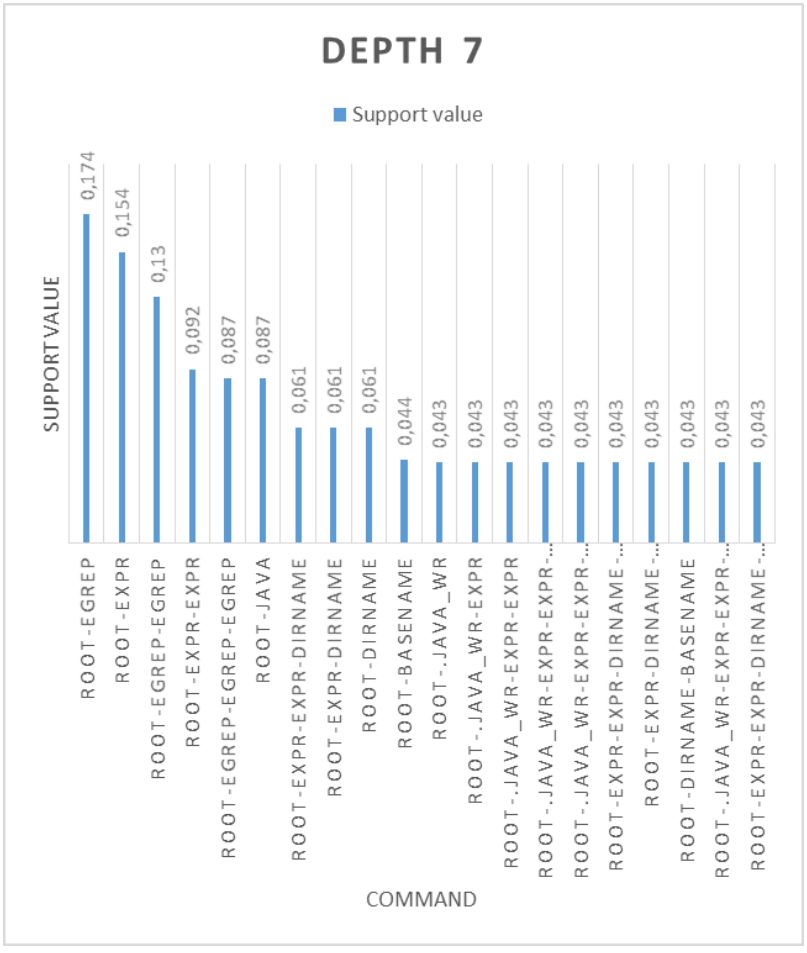

Ilustración 78: Graph with support values for User 3 with Depth 7.

User 4

Depth 3	
Support value	Node
0,204	Root-sh
0,105	Root-more
0,099	Root-more-sh
0,089	Root-sh-more
0,089	Root-sh-more-sh
0,074	Root-more-sh-more
0,062	Root-csh
0,048	Root-generic
0,042	Root-sendmail
0,041	Root-cat
0,039	Root-MediaMai
0,034	Root-rm
0,026	Root-ls
0,023	Root-toolches
0,023	Root-sh-MediaMai
0,020	Root-cat-mail
0,020	Root-cat-mail-csh
0,020	Root-mail
0,020	Root-mail-csh
0,019	Root-date

Tabla 110: Support values for User 4 with Depth 3.

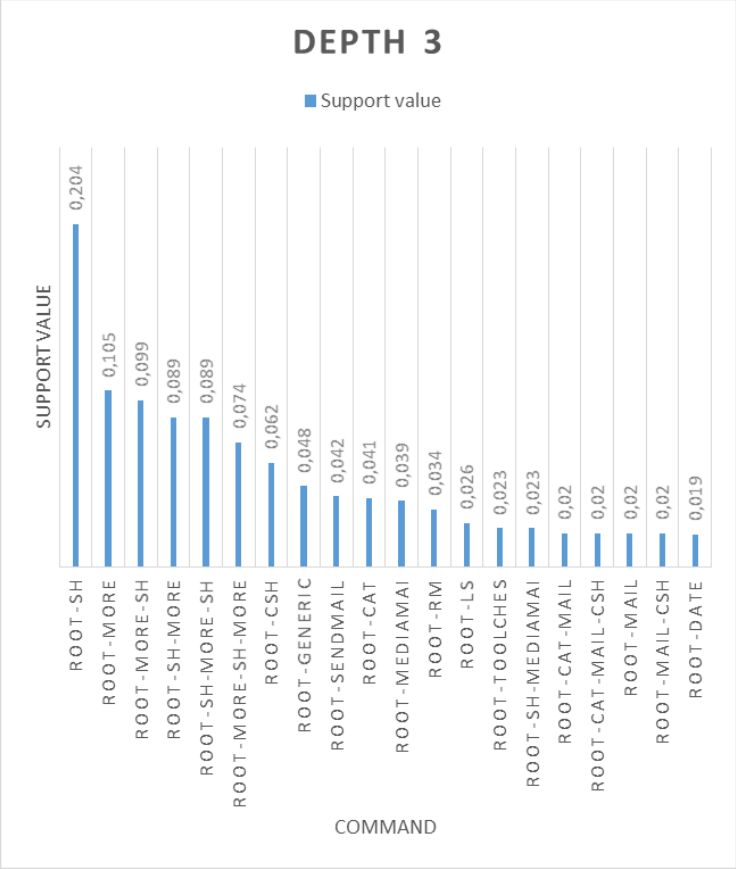

Ilustración 79: Graph with support values for User 4 with Depth 3.

Depth 5	
Support value	Node
0,204	Root-sh
0,105	Root-more
0,099	Root-more-sh
0,089	Root-sh-more
0,089	Root-sh-more-sh
0,074	Root-more-sh-more
0,074	Root-more-sh-more-sh
0,069	Root-sh-more-sh-more
0,069	Root-sh-more-sh-more-sh
0,062	Root-csh
0,059	Root-more-sh-more-sh-more
0,047	Root-generic
0,042	Root-sendmail
0,041	Root-cat
0,039	Root-MediaMai
0,034	Root-rm
0,026	Root-ls
0,023	Root-toolches
0,023	Root-sh-MediaMai
0,020	Root-cat-mail

Tabla 111: Support values for User 4 with Depth 5.

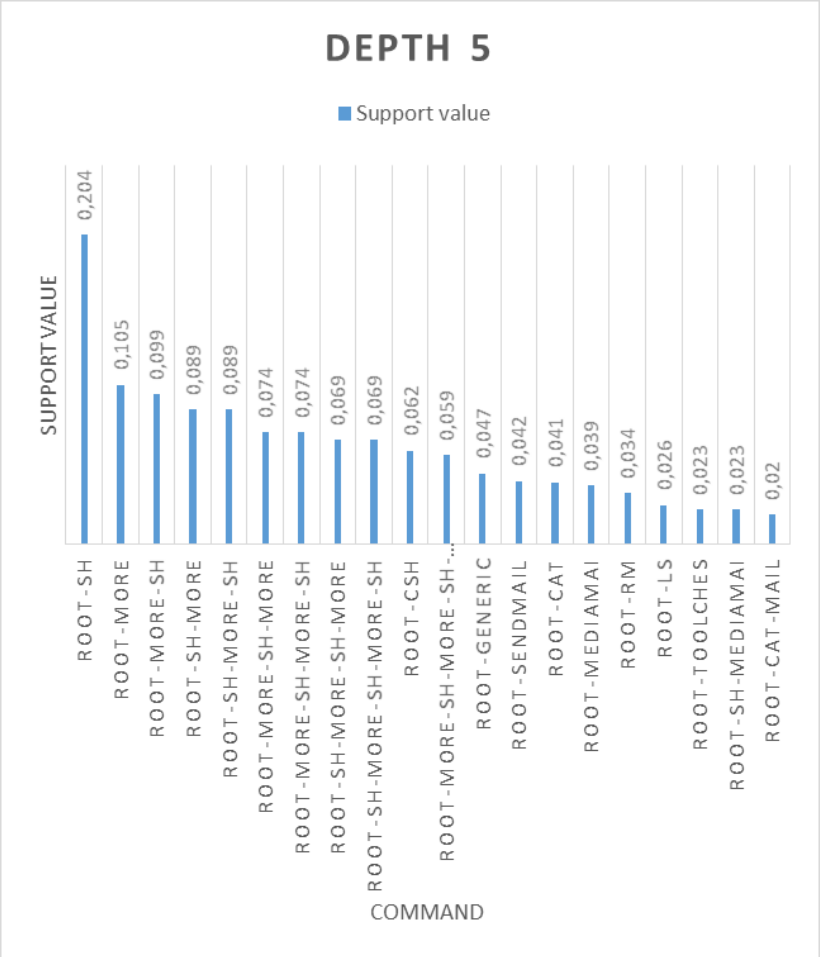

Ilustración 80: Graph with support values for User 4 with Depth 5.

Depth 7	
Support value	Node
0,204	Root-sh
0,105	Root-more
0,099	Root-more-sh
0,089	Root-sh-more
0,089	Root-sh-more-sh
0,074	Root-more-sh-more
0,074	Root-more-sh-more-sh
0,069	Root-sh-more-sh-more
0,069	Root-sh-more-sh-more-sh
0,062	Root-csh
0,059	Root-more-sh-more-sh-more
0,059	Root-more-sh-more-sh-more-sh
0,056	Root-sh-more-sh-more-sh-more
0,056	Root-sh-more-sh-more-sh-more-sh
0,048	Root-more-sh-more-sh-more-sh-more
0,047	Root-generic
0,042	Root-sendmail
0,041	Root-cat
0,039	Root-MediaMai
0,034	Root-rm

Tabla 112: Support values for User 4 with Depth 7.

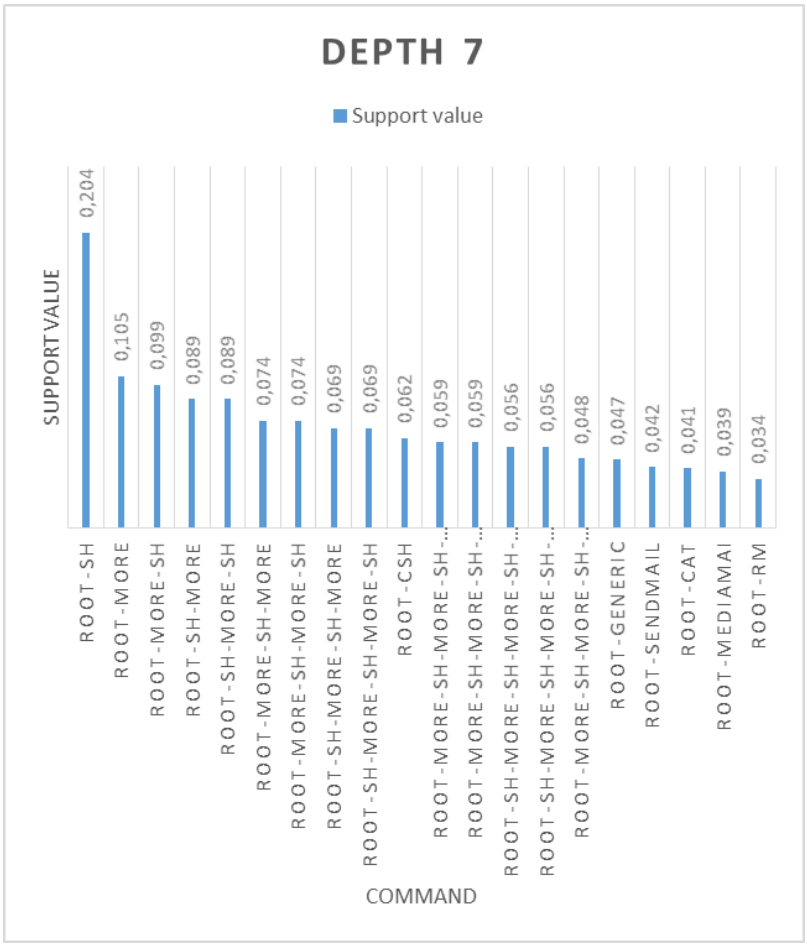

Ilustración 81: Graph with support values for User 4 with Depth 7.

User 5

Depth 3	
Support value	Node
0,112	Root-generic
0,07	Root-sh
0,05	Root-date
0,049	Root-cat
0,044	Root-true
0,04	Root-tcpostio
0,035	Root-ls
0,034	Root-rm
0,032	Root-date-generic
0,03	Root-ln
0,03	Root-tcpostio-tcpostio
0,027	Root-sed
0,024	Root-p
0,024	Root-p-sh
0,024	Root-lp
0,022	Root-LOCK
0,022	Root-ls-sed
0,022	Root-ls-sed-FIFO
0,022	Root-sed-FIFO
0,022	Root-FIFO

Tabla 113: Support values for User 5 with Depth 3.

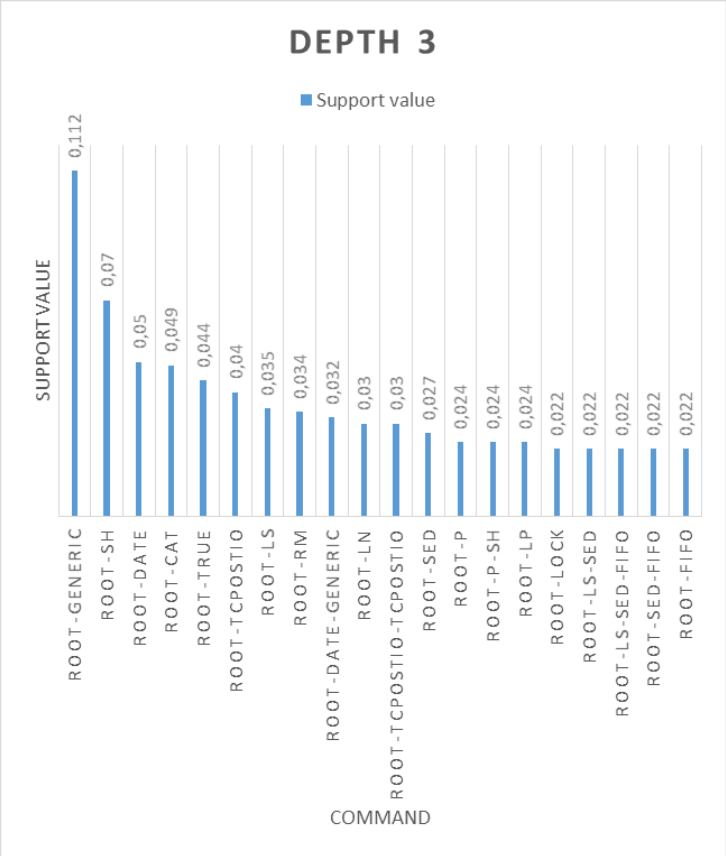

Ilustración 82: Graph with support values for User 5 with Depth 3.

Depth 5	
Support value	Node
0,112	Root-generic
0,07	Root-sh
0,05	Root-date
0,049	Root-cat
0,044	Root-true
0,04	Root-tcpostio
0,035	Root-ls
0,034	Root-rm
0,032	Root-date-generic
0,03	Root-ln
0,03	Root-tcpostio-tcpostio
0,027	Root-sed
0,024	Root-p
0,024	Root-p-sh
0,024	Root-lp
0,022	Root-LOCK
0,022	Root-ls-sed
0,022	Root-ls-sed-FIFO
0,022	Root-sed-FIFO
0,022	Root-FIFO

Tabla 114: Support values for User 5 with Depth 5.

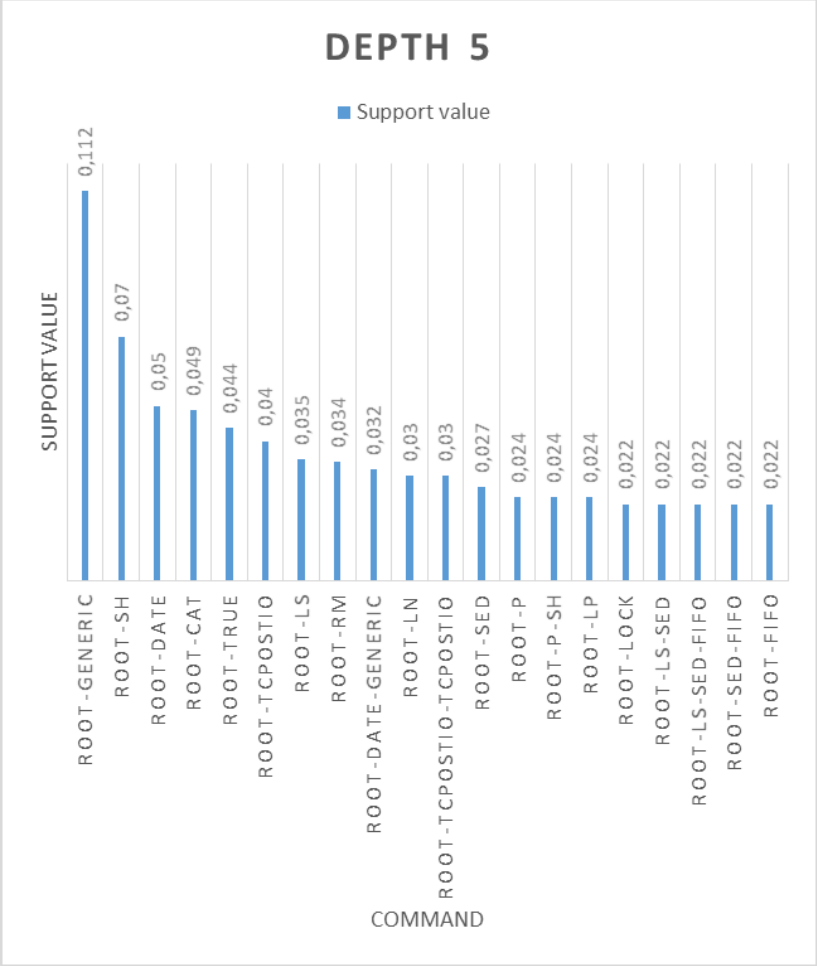

Ilustración 83: Graph with support values for User 5 with Depth 5.

Depth 7	
Support value	Node
0,112	Root-generic
0,07	Root-sh
0,05	Root-date
0,049	Root-cat
0,044	Root-true
0,04	Root-tcpostio
0,035	Root-ls
0,034	Root-rm
0,032	Root-date-generic
0,03	Root-ln
0,03	Root-tcpostio-tcpostio
0,027	Root-sed
0,024	Root-p
0,024	Root-p-sh
0,024	Root-lp
0,022	Root-LOCK
0,022	Root-ls-sed
0,022	Root-ls-sed-FIFO
0,022	Root-sed-FIFO
0,022	Root-FIFO

Tabla 115: Support values for User 5 with Depth 7.

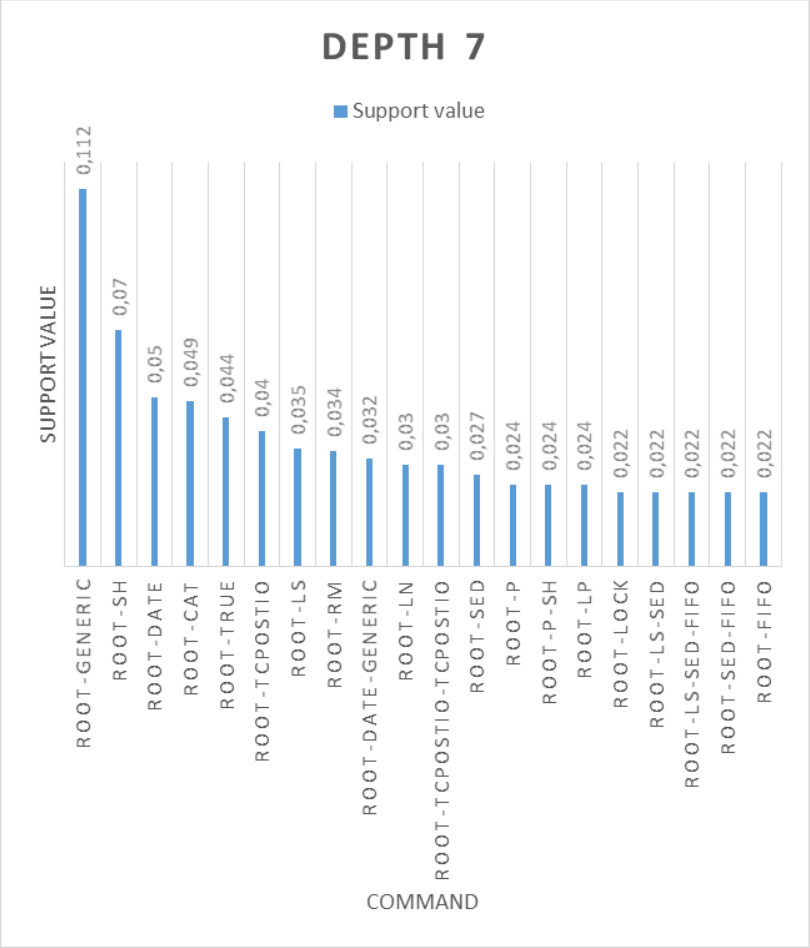

Ilustración 84: Graph with support values for User 5 with Depth 7.

2. Evaluation of the results

After performing the test battery, the obtained results can be interpreted as follows: as the support values measure the relevance of a sequence in a certain depth level, the most relevant commands for each depth can be extracted. The next table presents the most relevant commands for each user in the first three depth levels:

User	Depth	Most relevant command	Support value
1	1	netscape	0,278
	2	netscape-netscape	0,252
	3	netscape-netscape-netscape	0,230
2	1	gcc	0,138
	2	uname-nawk	0,069
	3	uname-nawk-cpp	0,053
3	1	egrep	0,174
	2	egrep-egrep	0,130
	3	egrep-egrep-egrep	0,087
4	1	sh	0,204
	2	more-sh	0,099
	3	sh-more-sh	0,089
5	1	generic	0,112
	2	date-generic	0,032
	3	ls-sed-FIFO	0,022

As show the results before, with the change of depth, the most relevant commands are not composed by the most relevant commands in the previous depth. This can be seen in the users 2, 4 and 5, where in level 2 of depth, the most relevant previous command is different that the command obtained in level 1. For example, taking as reference user 2, the most relevant command in depth 2 is formed by the sequence *uname-nawk*, while the most relevant command in depth 1 is *gcc*. For this reason, several levels of depth provide useful information about the studied users. These results show that is not necessary to select always the same level of segmentation depth for a specific case.

Users 1 and 3 show that the sequence of most relevant commands can be followed across the different levels of depth. For example, the most relevant command for user 1 is *netscape*, and following most relevant commands in depth 2 and 3 are composed by the most relevant commands in the previous levels of depth.

On the other side, the five users do not share similitudes regarding the use of the UNIX command terminal, as their most relevant commands are not the same. This shows that users have been using the command terminal with different purposes.

At last, support values for the most relevant commands are very low. This is caused by the large amount of commands in the same depth level. As remembrance, support values are calculated like:

$$support(x_i) = \frac{\text{insertionFrequency}(x_i)}{\sum_1^n \text{insertionFrequency}(x_{n,i})}$$

, where *i* represents the depth level and n represents each node of that depth level. If the denominator of the equation is too large regard the numerator, the support value tends to 0. This means that there is a large number of commands with the same length that users introduce in the UNIX command terminal.

Anexo V: Conclusions and future work

1. Conclusions

The realization of this project has resulted in the implementation of a tool capable of analyzing sequences of events regardless the chosen data domain. The main goal is been achieved with its realization.

Regarding the specific goals, they all have been also achieved.

With the implementation of the graphical user interface, the tool has a friendly, comfortable and functional user interface. The metrics with which the tool is equipped, allow to model an agent considering its most relevant actions. Finally, thanks to the export of trie in graphical format and applying colors on the nodes and links of the trie, the analyst is able to detect nodes of interest or sequences of actions relevant to the naked eye.

With the completion of the tool, the work of the analyst in charge of analyzing and studying sequences of events or actions is greatly lightened, because the process that makes the tool was done manually before.

With this said, the realization of the project is considered as successful.

2. Future work

Regarding future work that may serve to extend the functionality and performance of the tool implemented, the following works are considered:

- Parallelize the system's backend process. The parallelization of the process that calculates all the parameters of the trie and controls its creation would represent an increase in the performance of the tool. Times for the trie creation would be lower, so its use will still save more of the analyst's time who uses the tool.
- To be able to change the interface language. The graphical user interface should provide various languages, in order to help those who are not familiar with English to use the tool properly.
- Include as a method of segmentation the criteria of timing between events. Including this segmentation method will extend the functionality of the tool. With this, a more complete study of the sequence of events could be made, as the timing between events is a criterion to be considered in its interpretation.
- Include Chi-square as a metric. As this metric of relevance measurement is presented in the Doctoral Thesis *Modelado Automático del Comportamiento de Agentes Inteligentes* [62], the inclusion of this metric would expand the criteria for measuring the relevance of events in a sequence of events. Agent modelling would take another statistical approach.